REVISTA
DE TECNOLOGÍA E INFORMÁTICA
HISTÓRICA

Volumen 4 Número 1 2014

ARTÍCULOS




ISSN 2250-5431

Revista de Tecnología e Informática Histórica
Volumen 4, Número 1, 2014

Diego G. Macchi
Editor responsable

Revista de Tecnología e Informática Histórica es una publicación periódica propiedad de la Fundación Museo ICATEC. Museo de Informática, Computadoras y Accesorios Tecnológicos.
Tucumán 810 1er. Subsuelo, CP C1049AAR, Capital Federal.

# LA COMPUTADORA DE LAS MISIONES APOLO
# PARTE I: LA NECESIDAD DE UNA COMPUTADORA PARA IR A LA LUNA

Marcos Leguizamón*

En cada expedición a la Luna del programa espacial Apolo junto a los astronautas viajaba una computadora que cumplía un rol vital en la misión y que marcó un hito muy importante en la historia de la computación. Si pensamos desde el punto de vista de la tecnología actual, resulta extraño no incluir una computadora en cualquier proyecto tecnológico, desde los más avanzados a los más básicos. Las cámaras de fotos, los teléfonos, los autos, los aviones o los reproductores de música, por dar algunos ejemplos, tienen incorporadas una o varias computadoras para simplificar su utilización. Incluso sistemas electrónicos muy básicos, como un timbre, pueden contener varias computadoras pequeñas conocidas como microcontroladores o sistemas embebidos.

Si en cambio nos ubicamos a principios de la década de 1960, cuando se estaba planificando el programa Apolo para explorar la Luna, la idea de agregarle una computadora a la nave que iba a hacer ese trayecto no era para nada obvia, y, de hecho, si tenemos en cuenta el tamaño, peso y complejidad de operación de las computadoras disponibles en aquel momento, parecería una tarea imposible.

## LOS DESAFIOS DEL VIAJE A LA LUNA

La meta de viajar a la Luna se estableció en 1961 [1] a pocos días de que la Unión Soviética y luego los Estados Unidos llevaran a los primeros hombres al espacio. Hacía falta mucha experiencia y avances importantes en diversas áreas para poder llevar a cabo el programa [2], desde crear cohetes más confiables y potentes, mejorar los sistemas de comunicación y las cámaras de video, diseñar naves que soportaran varios tripulantes y que fueran mucho más maniobrables, desarrollar trajes espaciales para que los astronautas pudieran moverse en el vacío y también estudiar técnicas que hicieran posible que dos naves siguiendo trayectorias independientes se pudieran unir en el espacio, por enumerar algunas de las muchas innovaciones necesarias. Un aspecto al que se le dio mucha importancia

* RTI Latina S.R.L., leguizamon@gmail.com

desde el principio era el sistema de navegación a utilizar para guiarse en un trayecto tan largo (Figura 1).

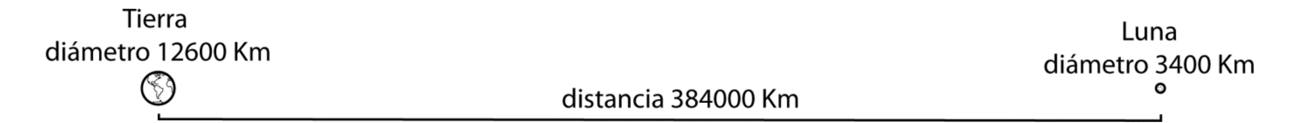



**Figura 1.** La Tierra y la Luna en escala. La distancia de 380.000 kilómetros que separan a los dos cuerpos celestes sigue siendo al día de hoy la mayor que un ser humano haya viajado.

El principal factor limitante en las misiones espaciales es el combustible. Los combustibles químicos tienen un peso y volumen que es importante para tener en cuenta, sobre todo porque al viajar al espacio es necesario partir con todo el combustible necesario para todas las etapas, ya que no hay como reabastecerse una vez que se inició el lanzamiento. En el momento del despegue el combustible representa la mayor parte del peso de un cohete. Por ejemplo, el satélite Sputnik pesaba unos 85 kilogramos pero el cohete que lo puso en órbita superaba las 270 toneladas, de las cuales unas 240 toneladas eran de combustible. Esto se debe a que la mayor parte del trabajo se utiliza para poder cargar el mismo peso del combustible y de ahí que por cada kilo de carga que se planee llevar al espacio, dependiendo del tipo de cohete, se necesitarán varios kilos de combustible extra. Por supuesto, hay un límite de peso a llevar porque una vez que se llena todo el tanque, es necesario agrandar el cohete, darle mayor potencia, y cambiar todos los parámetros.

La necesidad de no excederse con el consumo de energía hace que se planifique de forma tal que las maniobras a realizar sean óptimas y que se apliquen solamente los impulsos mínimos necesarios. Esto lleva a que sea impensado realizar una misión espacial controlando el viaje “a ojo” porque no es eficiente. Uno no puede pasarse de largo para analizar el camino y luego retomar, porque implicaría llevar mucho combustible extra desde el principio de la misión. A esto se suma que la percepción humana no es útil en un ambiente como el espacio, donde la determinación de distancias y velocidades es imposible sin instrumentos por la escala de las mismas. Si tomamos los instrumentos más simples y directos, como barómetros para medir la presión y determinar la altitud y velocidad del aire o las brújulas para la orientación, ninguno tiene utilidad en un viaje interplanetario.

En el espacio no hay referencias que se puedan usar para guiarse a simple vista con la precisión necesaria, e incluso llegando a la Luna la situación no mejora: el paisaje lunar es muy distinto al terrestre y no hay puntos de referencia fáciles de interpretar, no hay pistas de aterrizaje ni señalizaciones, pero tampoco ríos o vegetación que ayuden a determinar la escala, ni hay variedad de colores para guiarse. La falta de una atmósfera hace que a lo lejos se vea con la misma nitidez que de cerca porque no está presente el efecto atenuador del aire y esto también contribuye a la dificultad para distinguir las distancias visualmente.

## Los sistemas de navegación de vuelos de la época

A principios de la década de 1960, Estados Unidos y la Unión Soviética estaban en plena Guerra Fría con una carrera armamentística muy intensa. Se había avanzado mucho en sistemas de navegación, tanto para aviones como para los misiles, los cuales se estaban convirtiendo en el vehículo preferido para llevar las bombas atómicas al territorio enemigo [3]. Los Estados Unidos tenían en esa época varios proyectos independientes de misiles de largo alcance a cargo de distintas ramas de las fuerzas armadas y con diversas empresas e instituciones trabajando en ellos. Algunos de los proyectos eran el misil Redstone del ejército, la fuerza aérea tenía el Thor, el Atlas, el Minuteman y el Skybolt, y la armada el misil Polaris que se lanzaba desde un submarino. Todos estos proyectos de misiles con cabezas nucleares tenían características distintivas, como el tipo de plataforma de lanzamiento, tipo de combustible, el alcance y la capacidad de carga, pero algo que compartían era la necesidad de dar en el blanco y no detonar una bomba nuclear sobre un lugar equivocado.

El misil debía saber las coordenadas donde se encontraba en cada momento, la distancia y dirección hacia el objetivo para poder realizar las maniobras y llegar hasta allí. Se hicieron sistemas que controlaban al misil por radio desde bases en tierra, pero, a causa del uso militar, se avanzó para hacerlos completamente autónomos una vez lanzados y así ser inmunes a las interferencias del enemigo. Los acelerómetros se comenzaron a utilizar para medir desde adentro todas las aceleraciones sobre los misiles y la información obtenida se usaba para calcular la velocidad y posición que luego servían para guiar al misil por medio de distintos tipos de tecnologías. La información de los sensores era procesada por sistemas electromecánicos y con el tiempo también se fueron incluyendo más electrónica con válvulas o transistores, hasta que finalmente se empezó a aplicar electrónica digital con circuitos integrados, pero para aquella época todavía no se había llegado a utilizar una computadora corriendo un programa que dirigiera al misil.

La exploración espacial desde el principio estuvo muy vinculada con el área militar, de hecho los primeros cohetes que pusieron satélites y astronautas en el espacio eran versiones modificadas de los enormes misiles intercontinentales, diseñados para llevar una pesada bomba atómica a otro punto del planeta saliendo de la atmósfera terrestre en el camino y realizando una reentrada luego. Por supuesto que los sistemas de navegación de los misiles seguían siendo válidos para varias fases de una misión de exploración, pero en un viaje tan complejo como de la Tierra a la Luna esos sistemas solamente cubrían una fracción muy pequeña de las maniobras necesarias.

## Control humano de un viaje espacial

Las primeras naves espaciales eran muy simples: la Vostok soviética y la Mercury estadounidense eran cápsulas en las que apenas entraba una sola persona y estaban mayormente controladas desde tierra por radio. De hecho, antes de enviarse las primeras misiones tripuladas se probaba el correcto funcionamiento automático controlando todo

desde tierra y como tripulantes se llevaban animales (perros o monos), por lo cual era posible realizar la misión completa sin instrucciones del piloto.

Yuri Gagarin, el primer hombre en el espacio, no tenía control alguno de su nave a menos que en caso de emergencia abriera un candado con combinación que le permitiría tomar el mando [3]. Inicialmente se pensó que la clave del candado se le pasaría por radio si desde la base en tierra se aprobaba, pero se terminó dejando la clave anotada en un sobre dentro de la nave.

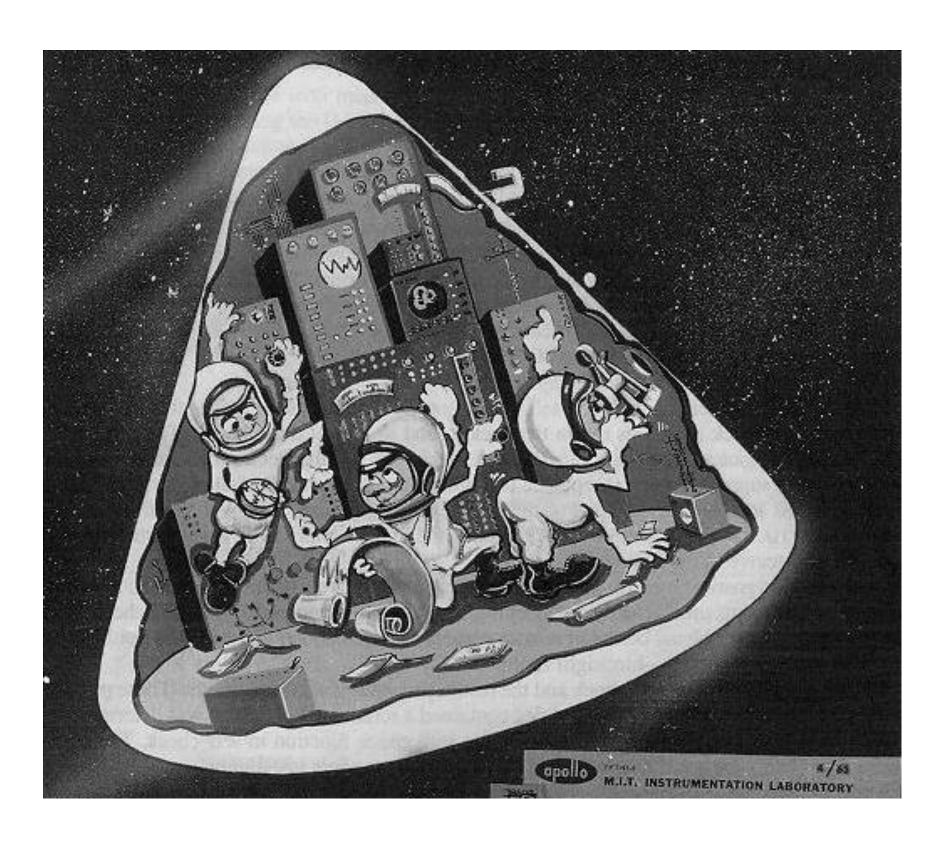
Las naves eran controladas por secuenciadores electromecánicos que, según instrucciones recibidas desde tierra por radiocontrol, disparaban una serie de acciones predefinidas y sincronizadas [4]. Desde la base también se recibían por radio automáticamente todos los parámetros de los sistemas de la nave y esta información, o telemetría, era visualizada en la central de control de vuelo donde se evaluaba.

En cuanto a la navegación, el astronauta cumplía un rol más bien de supervisor del funcionamiento general de la nave [3] y actuaba como sistema de backup en caso de detectarse fallas en los mecanismos automáticos. Desde el primer programa espacial de la NASA, el proyecto Mercury, se recurrió a los pilotos de prueba para ocupar los puestos de astronautas. Los pilotos de prueba se destacaban por ser pilotos hábiles y muy audaces que arriesgaban su vida probando aviones experimentales, pero además de estas habilidades también tenían un gran conocimiento teórico de ingeniería porque parte de su trabajo era hablar de igual a igual con los diseñadores de los aviones. Si bien la valentía, la aptitud física, la capacidad de tomar decisiones difíciles rápidamente y el conocimiento de ingeniería eran características útiles para un astronauta, muchos disentían en recurrir a este tipo de pilotos por que los veían muy individualistas y con la costumbre de correr riesgos innecesarios para demostrar su destreza. A lo largo de los programas espaciales hubo constantes choques entre los pilotos y los ingenieros para hacer las naves menos automáticas y darle más protagonismo al piloto, e incluso existieron propuestas para que los astronautas dirigieran manualmente los cohetes en el despegue, cosa que nunca ocurrió. Para el público los astronautas eran superestrellas y la cobertura en la prensa siempre destacó su habilidad para pilotear las naves. Esta situación minimizaba la importancia de los sistemas automáticos, pero la realidad es que la automatización era muy grande porque la complejidad de una misión espacial así lo requería (Figura 2).

## Computadoras analógicas y asistentes de vuelo

Los cohetes siempre necesitaron un sistema automatizado de estabilización porque a las velocidades que alcanzan no hay posibilidad de ser controlados por humanos, ni desde el interior ni a control remoto. En vuelo atravesando el aire a velocidades altísimas, el cohete debe realizar constantes ajustes para mantenerse bien orientado, y una pequeña desviación no compensada puede hacerle perder el control completo en una fracción de segundo. Por eso desde el primer misil, el V2 alemán en la Segunda Guerra Mundial, se incluyeron

sistemas con giroscopios conectados con los propulsores para mantener la dirección estabilizada [5].



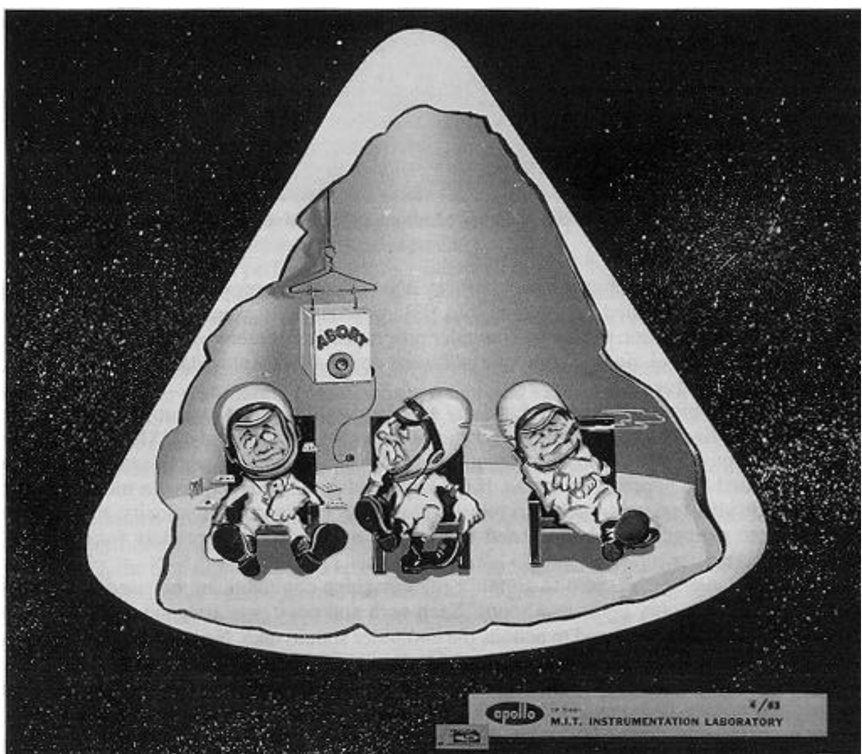




**Figura 2.** Caricaturas realizadas en la década del ‘60 en el MIT. En la primera una nave en donde los astronautas tienen que estar controlando cada pequeño detalle del vuelo. A continuación una nave con mucha automatización y astronautas ociosos. (Archivo NASA).

En el caso de los aviones, una característica de su diseño es su estabilidad o maniobrabilidad [3]: si es muy estable tiende a mantener su curso sin necesidad de constantes ajustes del piloto, pero esa estabilidad lo hace menos maniobrable y si el piloto deseara realizar una acción evasiva brusca, por ejemplo en un avión caza durante un combate aéreo, la estabilidad del avión se opondrá al movimiento y reaccionará de forma lenta. Un avión menos estable responde inmediatamente a los comandos del piloto, pero lo que es una ventaja en maniobras bruscas es una contra durante el resto del vuelo porque el piloto debe estar atento constantemente a los controles del avión para no perder sustentación. Una forma de obtener mayor maniobrabilidad y a la vez hacer al avión estable es utilizar un sistema intermediario entre el piloto y los controles del avión, que automatice operaciones repetitivas. De esta forma se construye un avión muy maniobrable pero poco estable y de la estabilización se encarga el sistema automático.

En la actualidad se conoce como *fly-by-wire* los mandos de aviones en donde el piloto controla al avión por medio de una computadora. Si no se tiene este tipo de control, el mando del piloto está físicamente conectado, por medio de poleas o sistemas hidráulicos, con los alerones y demás superficies de control de vuelo del avión. Con *fly-by-wire*, en cambio, el piloto le da instrucciones a la computadora y es la computadora la que, por ejemplo, se encarga de mover las superficies de las alas de forma tal de realizar las maniobras requeridas o mantener la dirección actual.

En la década del 1950 no se tenía la complejidad tecnológica actual, pero se comenzaron a usar computadoras analógicas como asistentes de vuelo [3]. Una computadora analógica no era una computadora en el sentido que podemos entenderlo hoy, sino mecanismos que realizaban cuentas y ciertas operaciones automatizadas, pero no ejecutaban un programa (software) y obviamente no eran digitales porque no trabajaban con números bien definidos sino con valores continuos y los componentes electrónicos de estas computadoras eran, por ejemplo, amplificadores y filtros.

El avión cohete X-15 fue un salto muy importante en el uso de computadoras analógicas como asistentes de vuelo [3]. En este proyecto iniciado a fines de la década del '50 por NACA, la agencia aeroespacial estadounidense que luego fue reemplazada por la NASA, el pequeño avión despegaba desde otro avión más grande (un bombardero B-52), llegaba hasta una altura de aproximadamente 100 kilómetros, y abandonaba la atmósfera volando en el espacio pero sin entrar en órbita. Luego debía realizar la reentrada a la atmósfera y aterrizar en una pista en tierra.

Controlar este avión era muy complejo porque no estaba diseñado para un solo tipo de operación, sino para moverse en condiciones muy variadas, por ejemplo en la atmósfera a velocidades récord de Mach 6,7, maniobrar en el vacío del espacio, realizar una reentrada a la atmósfera y luego aterrizar en una pista a velocidades subsónicas. Su diseño lo hacía poco estable en la atmósfera, su velocidad tan alta no perdonaba ninguna distracción y en el vacío necesitaba usar unos pequeños propulsores extra para orientarlo. Por estas razones, se utilizaba una computadora analógica que manejaba al avión. El piloto por medio de los

controles de vuelo le decía a la computadora lo que deseaba hacer y la computadora aplicaba el estado deseado sobre las alas y los propulsores del avión y lo mantenía estable.

**Teoría de control**

El avance de la electrónica en la década de 1950 hizo que esta empezara a ser utilizada en sistemas para controlar procesos físicos, como en el caso del control de vuelo del X-15. Los principios teóricos para cumplir esa función se venían estudiando desde el siglo XIX y actualmente se la conoce como teoría de control. Esta es una rama de la ingeniería que estudia el control de procesos dinámicos, ya sea usando medios mecánicos, electrónicos o por software. Una forma de clasificar a un sistema de control es si es de lazo cerrado o abierto.

Un sistema de control de lazo abierto envía una señal al proceso que desea controlar y confía que el resultado a obtener sea el esperado. Un ejemplo de un sistema de este tipo es una tostadora de pan con control por tiempo, que aplica calor al pan durante un tiempo prefijado estimando que ese tiempo es el necesario para tostar al pan, pero cuando las condiciones no son las ideales, por ejemplo el pan ya estaba medio tostado de antes o la resistencia no calienta lo suficiente porque ya tiene mucho uso, en lugar de aplicar el calor durante menos tiempo en el primer caso y más tiempo en el segundo, intentando siempre calentar al pan hasta un nivel adecuado, se limita a sacar el pan en el tiempo prefijado creyendo ciegamente que lo que hace está siempre bien.

En un sistema de lazo cerrado se utiliza el principio de feedback (retroalimentación en castellano) para ajustar la señal de control en base al estado en que está el proceso físico que se controla. Se lee el estado en que se encuentra el proceso y la diferencia entre lo leído y el estado deseado se convierte en la señal de control y de esta forma se puede mantener el resultado estabilizado. Este tipo de feedback en donde se resta la diferencia se llama feedback negativo. El positivo, en cambio, es generalmente un efecto indeseable, un círculo vicioso, en que la diferencia entre el modelo ideal y el estado real se hace cada vez más grande.

Un ejemplo simple es un controlador de velocidad crucero de un auto (Figura 3), en que el sistema tiene una velocidad a la que se desea viajar. Se obtiene el feedback sensando la velocidad actual del vehículo: si está andando más rápido entonces el sistema presiona el freno y baja la potencia del motor, si está andando más lento lo que hace es subir la potencia del motor. Realizando esto constantemente se puede mantener la velocidad estable sin realmente conocer las condiciones en que se mueve el vehículo: si está en subida o en bajada, si es un camino asfaltado o de tierra o siquiera el peso que lleva. De todas formas es deseable que el sistema conozca algo del funcionamiento del auto, que tenga un modelo teórico, porque si por ejemplo se desea ir a 100 km/h y se sensa que está andando a 99 km/h, debería saber aproximadamente cuánta más potencia darle al motor. Si simplemente se limita a darle un 50% más de potencia se va a pasar de largo y luego va a

tener que frenar también en demasía y otra vez se va a pasar, y quedará oscilando fuertemente alrededor de 100Km/h sin alcanzar una posición estable.

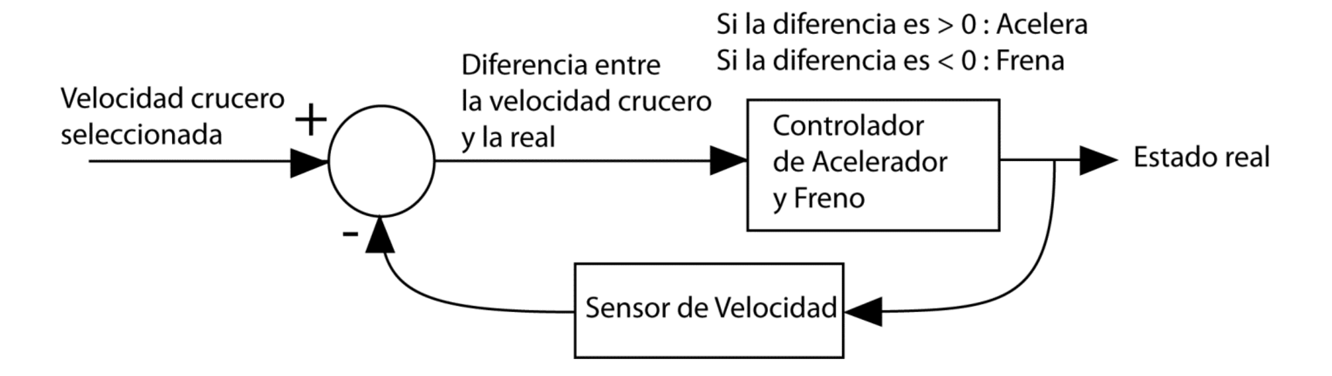



**Figura 3**. Sistema simple de control de velocidad crucero con feedback. Leyendo la velocidad actual se decide cómo actuar.

Desde la revolución industrial en el siglo XVIII se utilizan sistemas de control mecánicos que se valen de técnicas similares para regular la velocidad de motores de vapor y también son comunes sistemas de este tipo en, por ejemplo, un termostato de una estufa, en donde se estabiliza la temperatura a un valor seleccionado por la dilatación de unos metales que mecánicamente regulan el flujo de gas. Con la electrónica el rango de control a realizar utilizando feedback se amplió mucho porque con computadoras analógicas se puede modelar sistemas más complejos en menor espacio y con mayor facilidad.

El uso de computadoras digitales programables da mucho más poder para desarrollar un sistema de control, porque en un programa se pueden manejar con facilidad muchas más variables y factores, y así modelar y controlar el proceso físico con mayor precisión. La división de tareas al momento del diseño es mucho más eficiente, porque los encargados de la electrónica sólo deben proveer un sistema que le entregue a la computadora la información que necesita y realice las acciones que esta pida, sin conocer ni los algoritmos que se usarán ni el orden en que se ejecutaran. La gente encargada del software trabaja por su cuenta y si todo fue correctamente planificado, el programa correrá en el sistema final sin necesidad de realizar modificaciones a la electrónica.

Un programa de computadora que controla un sistema físico debe tener un conocimiento de los principales factores que rigen a ese sistema para poder controlarlo eficientemente y por medio del feedback se ayuda a regular los detalles menores. Es importante que el sistema detecte cuando la diferencia entre el modelo ideal y el estado actual es muy grande, porque así se pueden detectar errores. Por ejemplo, el sistema de velocidad crucero debería alertar al conductor cuando se está yendo mucho más lento de lo esperable, porque puede deberse a algún problema mecánico que conviene atender.

En el caso del sistema de control de una nave espacial interplanetaria, en donde hay varias etapas con trayectorias y maniobras muy distintas, varios propulsores con características propias, se requiere mucha precisión y hay muchas variables, como el peso que va reduciéndose a medida que se consume el combustible y el campo gravitatorio que

varía según la ubicación, el uso de una computadora programable es casi indispensable. Los principios siguen siendo los mismos pero la integración de múltiples lógicas se simplifica mucho.

## MISIÓN APOLO

Para entender la función que cumpliría la computadora en el viaje a la Luna, veamos una descripción aproximada de cómo se debía desarrollar cada misión Apolo [2].

La meta desde el principio consistía en enviar un cohete tripulado hasta la Luna, a 380.000 kilómetros de distancia de la Tierra, y posarse sobre su superficie. Como no hay atmósfera en la Luna, no se pueden realizar un descenso con paracaídas, por eso, para contrarrestar la gravedad lunar (6 veces menor a la terrestre), se debería utilizar un propulsor operando en la dirección opuesta para tocar la superficie a baja velocidad y no estrellarse. Una vez que los astronautas realizaron la exploración del terreno en la Luna, la nave debería despegar hacia la Tierra y volver a recorrer los 380.000 kilómetros de separación, entrar en la atmósfera terrestre y descender, ahora sí, utilizando paracaídas (Figura 4).

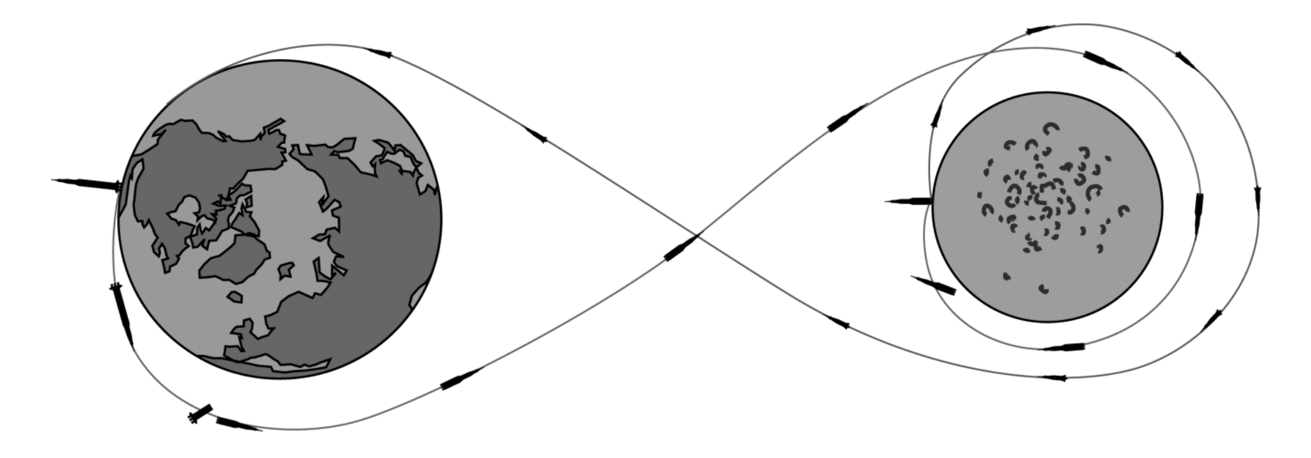


**Figura 4.** Esquema de viaje directo ida y vuelta a la Luna. Sin escala.

Para poder planificar la misión se debe analizar la potencia necesaria del cohete y de ahí su tamaño y peso. El problema de realizar la misión de forma directa es que se necesita depositar un cohete muy poderoso y pesado en la Luna para que sea capaz de partir desde allí hasta la Tierra, y todas las etapas previas debían cargar ese peso. El cohete que tendría que partir desde la Tierra para poder llevar a la Luna toda esa carga se llamaba Nova y nunca pasó de la etapa de diseño porque estaba claro, por sus dimensiones, que su construcción y utilización se encontraban muy por fuera del terreno de lo posible.

Lo que se propuso inicialmente era dividir la etapa que consume más energía, que es la de abandonar la Tierra, y enviar varios cohetes en lugar de sólo uno y ponerlos en órbita alrededor de la Tierra, y allí unirlos y formar un cohete poderoso que podría realizar el resto

del trayecto ida y vuelta hacia la Luna. Esta idea, llamada *Earth orbit rendezvouz* (encuentro en la órbita terrestre), seguía teniendo problemas de implementación, sobre todo porque había que construir muchos cohetes muy grandes y lanzarlos en un lapso breve de tiempo, y por lo general pasaban meses entre despegue y despegue por lo complejo de cada uno.

Un ingeniero de la NASA, John Houbolt, propuso un método alternativo que simplificaba la misión al realizarse con un solo despegue. Se partía desde la Tierra a la Luna y allí en lugar de alunizar se entraba en órbita. En esta etapa la misión se dividía en dos y una nave muy liviana, el Módulo Lunar, descendía sobre la Luna mientras la otra seguía en órbita. Luego el Módulo Lunar despegaba, pero no para ir hacia la Tierra, sino para volver a entrar en órbita y unirse otra vez a la nave principal, en un viaje que necesitaba mucha menos energía. La nave principal, con un propulsor más poderoso que el Modulo Lunar, y partiendo desde su órbita alrededor de la Luna, en lugar de hacerlo desde la superficie, abandonaba la Luna y volvía a la Tierra. Este plan de misión se llamaba *Lunar orbit rendezvous* (encuentro en órbita lunar) e inicialmente fue resistido por considerarse muy riesgoso realizar esas maniobras tan lejos de la Tierra, pero, con el tiempo, mientras más se estudiaban las distintas posibilidades se vio que era el único plan viable y fue el que terminó implementándose en las misiones Apolo.

**Naves y tripulantes**

La misión Apolo constaba de un cohete, dos naves, el Módulo de Comando, el Módulo Lunar y tres astronautas: el comandante de la misión, el piloto del Módulo de Comando y el piloto del Módulo Lunar (Figura 5).

**Cohete Saturno V**

Es el cohete de 3 etapas que envía las naves hacia la Luna. Cada una de las tres etapas que lo componen funciona por turnos, y cuando una etapa del cohete consume su combustible y concluye su función, se eyecta para ahorrar peso y se activa la siguiente. Este cohete de 110 metros de altura y 2800 toneladas de peso y capaz de enviar hacia la Luna 45 toneladas es el mayor jamás construido y la máquina más poderosa que alguna vez haya volado.

**Módulo de Comando y Módulo de Servicio**

En la punta del Saturno V se encuentra el Módulo de Comando, que es la nave principal donde están los tres astronautas la mayor parte del tiempo. El piloto del Módulo de Comando es el encargado de controlar esta nave y no participa del alunizaje sino que permanece orbitando la Luna mientras sus compañeros la exploran.

Esta nave está formada por dos módulos, el de comando y el de servicio, que está unido a él por detrás y no es accesible a los astronautas. El Módulo de Servicio lleva un

poderoso propulsor, necesario para entrar y salir de órbita en la Luna, hidrógeno y oxígeno para generar electricidad y para proveer a los astronautas de agua, entre otras cosas. Esta sección de la nave es eyectada antes de reingresar a la Tierra, para entrar lo más liviano posible y porque tapa el escudo térmico que protege al Módulo de Comando de las altas temperaturas del reingreso. El Módulo de Comando es la única nave que regresa a la Tierra luego de la misión, el resto se va descartando en distintas etapas.

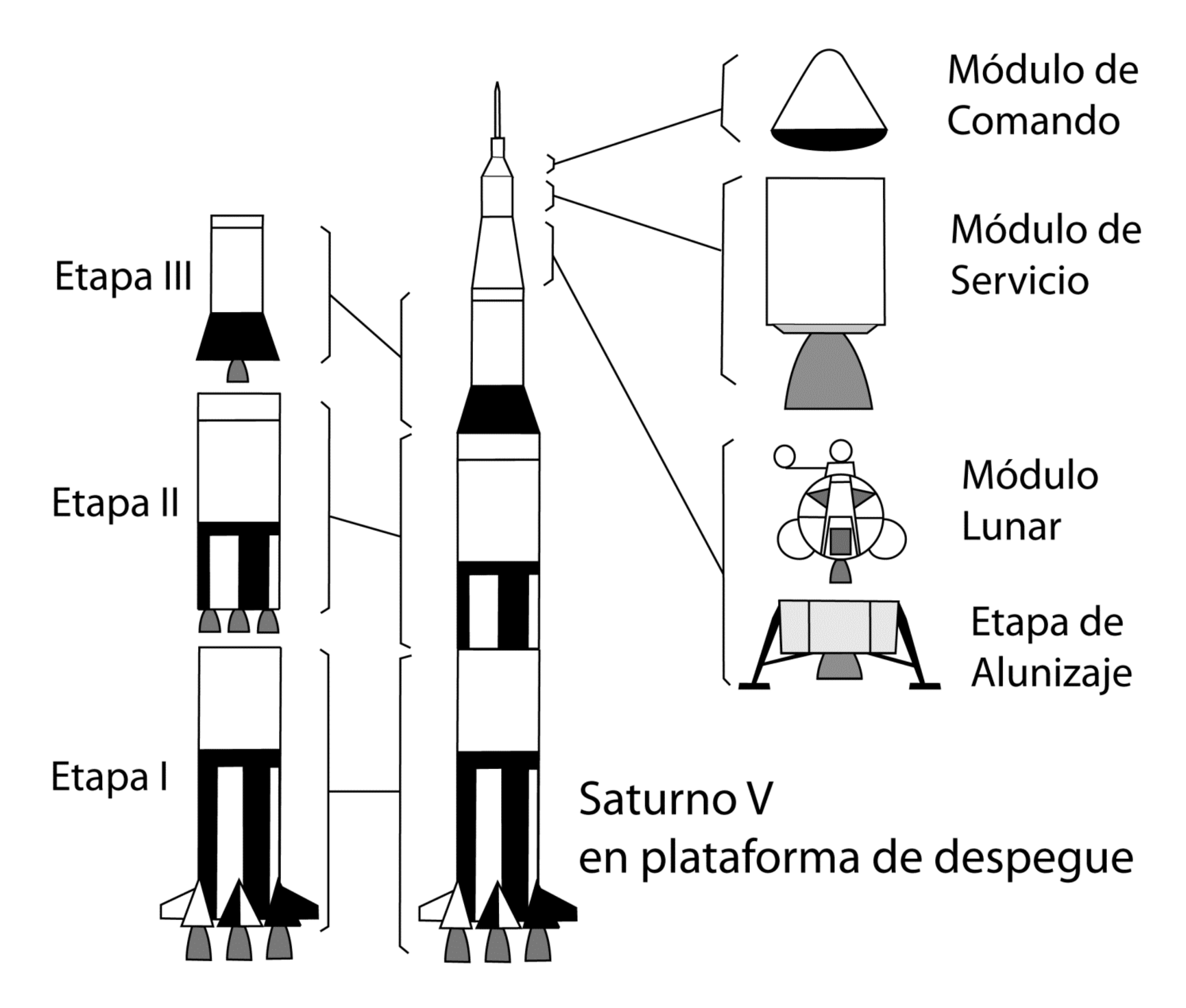



**Figura 5.** Naves y cohetes de la misión por separado y agrupados como lo estaban en la plataforma de despegue.

**Módulo Lunar**

Esta nave viaja inicialmente debajo del Módulo de Comando, dentro del Saturno V, y sólo sirve en el vacío del espacio y la Luna, porque, con la meta de ahorrar peso lo más posible, su estructura es extremadamente frágil. El comandante de la misión es el encargado de pilotear esta nave, no el piloto del Módulo Lunar, cuyo título se presta a confusiones y en realidad oficia de copiloto y encargado de sistemas. Este módulo a su vez se divide en dos al

despegar de la Luna, abandonando toda la parte inferior que se utiliza sólo para alunizar. Por esa razón tiene dos propulsores, uno para descender y otro para despegar.

**Descripción de una misión Apolo**

La misión [6] comienza con el despegue del cohete Saturno V, las dos primeras etapas se consumen y descartan para, 12 minutos después del despegue, poner a la tercera etapa junto con el Módulo de Comando y Módulo Lunar en órbita alrededor de la Tierra a 190 km de altura, dando una vuelta completa al planeta cada 90 minutos.

Esta órbita se llama órbita de estacionamiento y es un paso intermedio hacia la Luna. Aquí la tercera etapa del Saturno (la única que queda) permanece con el propulsor apagado pero aún con mucho combustible. En esta órbita temporal se espera unas 2 horas y, estando en el lugar adecuado, se vuelve a encender el motor del Saturno. Esta es la última vez en que se usa este cohete y este impulso, llamado *inyección translunar* (TLI por su sigla en inglés), es el que llevará a las naves directo hacia la Luna.

En camino a la Luna, un viaje de 380.000 km, antes de eyectar al Saturno se despliega el Módulo Lunar, que todavía estaba protegido en el cohete, y se lo une al Módulo de Comando por la trompa, y luego de este evento, ambas naves son accesibles a los astronautas.

Unos 3 días más tarde, estando ya cerca de la Luna, se utiliza el propulsor del Módulo de Servicio para frenar la nave hasta una velocidad que permita entrar en órbita lunar. La misión está diseñada para que, si no se encendiera este propulsor aquí, la trayectoria libre (sin que la nave aplique impulsos) lo lleve directo otra vez hacia la Tierra.

El Módulo de Comando y Módulo Lunar quedan orbitando la Luna a una altura de 110 km dando una vuelta completa a la Luna en aproximadamente 120 minutos. Ahora, el Módulo Lunar se desprende del Módulo de Comando con los astronautas en él, y con su propulsor modifica su órbita a una cuyo punto más bajo (periápside o perilunio para este caso) es de 16 km. En este punto de menor altura se vuelve a encender el propulsor de descenso y se comienza a aproximar a la superficie. El comandante selecciona el lugar más adecuado para posar la nave y comienza a descender verticalmente hasta tocar el suelo a baja velocidad (Figura 6).

Una vez concluida la exploración de la superficie, el Módulo Lunar despega. Lo hace utilizando un propulsor dedicado sólo a este fin, que no es el mismo que se utilizó para descender, y la mitad inferior del módulo, que incluye el propulsor de descenso, queda sobre la Luna para ahorrar peso. El Módulo Lunar entra en órbita y, por medio de maniobras orbitales y con un radar dedicado, busca al Módulo de Comando para iniciar el procedimiento de acople. Una vez unidos, los tres astronautas quedan en el Módulo de Comando y se desprenden del Módulo Lunar que ya no cumple ninguna función y es un peso innecesario.

Estando en la posición adecuada, el Módulo de Comando enciende su propulsor principal y abandona la órbita lunar en dirección hacia la Tierra (inyección transterrestre).

Luego de recorrer otra vez los 380.000 km de distancia, esta vez el viaje tendrá una duración de sólo unas 60 horas porque se está cayendo sobre el cuerpo de mayor gravedad de los dos. Esto implica una mayor aceleración y velocidad, y estando a corta distancia de la Tierra a la máxima velocidad (unos 11 km por segundo), se desprenden del Módulo de Servicio y queda al descubierto el escudo térmico del Módulo de Comando.

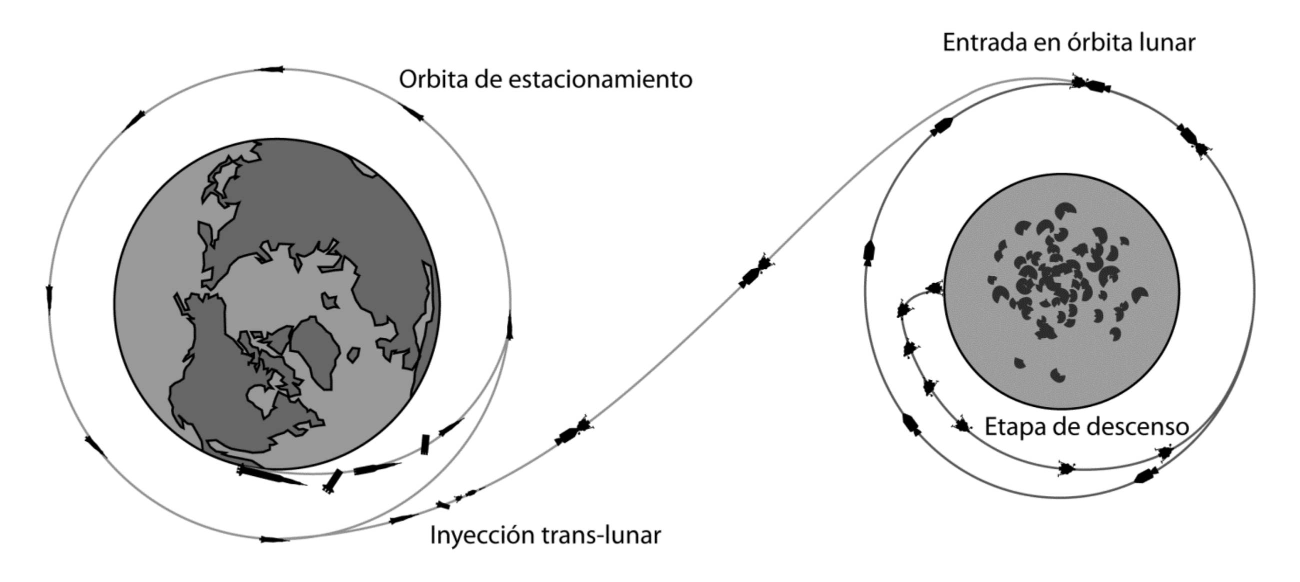



**Figura 6.** Diagrama de la misión desde el despegue al alunizaje

La nave se ubica con el escudo apuntado hacia adelante para enfrentar la atmósfera con él, que lo protege de las altas temperaturas por la fricción a tan alta velocidad, y recorre unos 2.000 km con una desaceleración muy alta que en su punto más intenso llega a 6 G (59 $m/s^2$). A diferencia de las primeras cápsulas espaciales, como la Vostok de Gagarin o las Mercury, que caían sin ningún control siguiendo una trayectoria balística, el Módulo de Comando controla el vuelo a través de la atmósfera a pesar de no tener alas [6] por medio de giros sobre su eje aprovechando que su centro de gravedad no está en el centro de la nave y por su forma recibe sustentación del aire, y así logra dirigirse hacia un punto prefijado sobre el océano. Ya estando a 7 km sobre el nivel del mar comienzan a desplegarse los paracaídas que hacen que la cápsula toque la superficie del océano a baja velocidad y luego allí los astronautas son auxiliados por helicópteros que los estaban esperando en la zona (Figura 7).

## EL NACIMIENTO DE LA COMPUTADORA DE NAVEGACIÓN DE APOLO

Dentro del Instituto de Tecnología de Massachussets (MIT) funcionaba, con cierta independencia de esta institución, el *laboratorio de instrumental*. Este laboratorio, desde la década del 30, venía dedicándose a la investigación y creación de instrumentos, principalmente para navegación aérea, siendo pioneros en el uso de giroscopios [3]. Desde el principio trabajaron con las fuerzas armadas para distintos desarrollos bélicos

aeronáuticos y también con sistemas de control de artillería durante la Segunda Guerra Mundial. Luego, con la entrada en escena de los misiles, participaron activamente en la creación de sistemas de navegación más modernos. Esta estrecha vinculación con la industria armamentística, incompatible con una institución educativa, hizo que en la década del 70 se independizara por completo del MIT y se convirtiera en el Laboratorio Draper, en honor a quien fuera su creador y director durante muchos años.

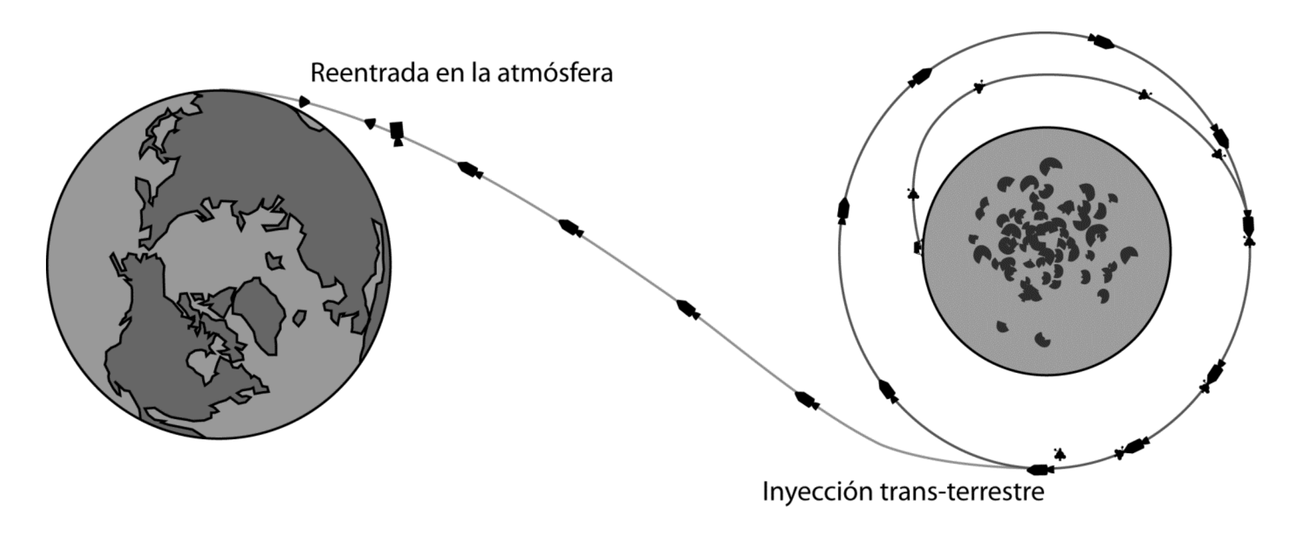



**Figura 7.** Diagrama de la misión desde que se abandona la superficie lunar hasta el regreso a la Tierra

Charles Stark Draper era un ingeniero con formación en psicología que lideró el campo de sistemas de navegación aéreos. En 1953 realizó un innovador sistema inercial que tomó el control de un enorme bombardero durante unos 5000 Km de Boston a Los Angeles, atravesando todo Estados Unidos de una punta a la otra sin control humano ni ninguna referencia externa [7].

También en la década del 1950 Draper y su equipo fueron los encargados de realizar los sistemas de navegación inercial de los misiles nucleares Thor y Polaris [3]. Este último tenía como principal reto ser lanzado desde un submarino sumergido, por eso debía ser muy liviano y pequeño, lo que hacía que el sistema de navegación, además de ser muy preciso, también estuviera limitado en volumen y peso. Eldon Hall, que estaba encargado en el laboratorio del estudio de la lógica digital, introdujo esta tecnología al sistema de navegación en lugar de incluir una computadora analógica. A pesar de ser un sistema digital, no era lo que hoy llamaríamos una computadora porque no corría un programa sino que eran circuitos con funciones fijas y cualquier modificación que se deseara debía hacerse modificando el circuito y no un software. El sistema digital estaba fabricado utilizando transistores individuales y eran necesarios varios componentes para armar cada compuerta lógica básica.

En 1957 mientras el laboratorio del MIT estaba involucrado en sus proyectos militares, la Unión Soviética puso en órbita el satélite Sputnik e inició la carrera espacial. En ese momento un grupo dentro del laboratorio empezó a planificar en papel un proyecto

para llevar una sonda a Marte, que una vez que estuviera lo suficientemente cerca sacaría un sola fotografía en celuloide y luego emprendería la vuelta a la Tierra con el film para ser revelado, en una expedición que duraría en total 3 años [3] [8]. El proyecto era liderado por Hal Laning, un matemático que había trabajado en distintas innovaciones en la computadora Whirlwind, la primera computadora creada por el MIT en la década del 40 y junto a él un físico especializado en óptica, Milt Tragenser. El diseño giraba alrededor de las especialidades del equipo: un sistema de navegación inercial para guiar a la nave y un sistema óptico para tomar como referencia las estrellas y los planetas para orientarse y calibrar al sistema inercial, todo coordinado por una computadora digital programable que controlaría el trayecto de forma autónoma. Al grupo que diseñaba el proyecto se incorporó el argentino Ramón Alonso [9] [10] que venía del laboratorio de computación de Harvard, junto con el matemático del MIT Richard Battin [8], un pionero en el estudio y diseño de trayectorias interplanetarias.

El informe detallado del proyecto de la misión a Marte lo presentaron a la recién creada NASA, pero fue rechazado por considerárselo muy poco práctico [3]. El Jet Propulsion Laboratory, en California, que sería la institución encargada de las misiones robóticas de exploración del sistema solar de la NASA, le daba un papel central a la comunicación por radio, logrando grandes avances en la comunicación interplanetaria que luego se usarían en las propias misiones Apolo a la Luna. La sonda del MIT prácticamente ignoraba las ondas de radio y debía realizar un viaje ida y vuelta para traer una foto en lugar de transmitirla y la navegación misma no aprovechaba el guiado desde la Tierra porque era completamente autónomo pero también tenía una electrónica más compleja que las primeras sondas. A pesar de eso, la NASA se mostró interesada en el diseño de la computadora y financió al grupo para que Laning y Alonso siguieran avanzando con este desarrollo electrónico.

En el año 1961, apenas dos meses después de fijada la meta de viajar a la Luna, cuando recién se estaba discutiendo que técnica usar para realizar la misión y antes de definir los fabricantes de los cohetes y naves espaciales, se le dio el contrato al MIT para crear la computadora de navegación de la misión Apolo. En ese momento incluso faltaban meses para realizar el primer vuelo orbital del proyecto Mercury. Este contrato fue el primero de todo el programa y el único que se adjudicó de forma directa sin llamar a licitación [3], lo cual no es un detalle menor existiendo empresas con mucha experiencia en diseño y fabricación de computadoras, como Honeywell, General Electric o RCA, pero principalmente IBM que durante mucho tiempo fue sinónimo de computación. Una organización sin fines de lucro se impuso sobre los pesos pesados de la industria que también tenían una experiencia importante en control de vuelo y navegación de otros sistemas armamentísticos. Si bien pudo haber razones políticas para la selección, la NASA no se centró tanto en la computadora sino en la habilidad para crear un sistema de navegación integrado donde la computadora era una parte del proyecto en sí.

El MIT se encargaría del diseño de la computadora y los instrumentos de navegación, ya sea del sistema inercial con sus giroscopios y acelerómetros, como del sextante para la calibración del sistema inercial y la consola para que los astronautas interactuaran con la

computadora. Además de diseñar todo el hardware también deberían ser los encargados de escribir el software que correría en la computadora.

El plan sería diseñar la computadora y crear una primera implementación, llamada Block I, que serviría para pruebas de vuelo porque, a pesar de incluir todas las características finales, estaría hecha con una tecnología intermedia y no cumpliría los requisitos de peso, consumo eléctrico, volumen, etc. A la vez que se diseñaba la computadora se investigaría sobre tecnologías de componentes electrónicos y armado de placas para buscar un equilibrio óptimo entre tamaño, peso y fiabilidad, y luego la computadora final que viajaría a la Luna seria la Block II. El cronograma optimista de la NASA en 1961 estimaba que el Block I debía estar para el 63 y el II un año después, todo con un presupuesto de 4 millones de dólares donde proyectaban que hacia el final del proyecto había en total 300 personas trabajando en él [3].

## CARACTERÍSTICAS DEL VUELO ESPACIAL

Para comprender el control que debía realizar la computadora sobre las naves Apolo primero es necesario conocer a grandes rasgos como se desplaza una nave en el espacio.

La característica principal del movimiento en el espacio exterior es que se hace en el vacío. En la Tierra la atmósfera que la rodea hace que cada movimiento sea frenado dependiendo de la velocidad que se lleve. A un auto, por ejemplo, el aire le opone fuerzas cada vez mayores a medida que alcanza velocidades más altas, por eso los vehículos que necesitan moverse más rápido adoptan formas aerodinámicas para minimizar estas fuerzas. En el vacío, en cambio, una vez que un objeto toma una velocidad, esta se mantiene y para alcanzar velocidades mayores lo único necesario es acelerar durante más tiempo: para el caso es lo mismo moverse a 10 km/h que a 10 km/s. Las leyes del movimiento y la inercia de los cuerpos son mucho más evidentes en el espacio exterior que en la Tierra donde los efectos del aire hacen que la inercia no pareciera existir. De hecho, recién en el siglo XVII se descubrió su existencia de la mano de Galileo y Newton, y las teorías anteriores que describían estos movimientos hoy nos resultan absurdas.

Cada nave espacial cuenta con un motor principal que aplica una fuerza y gana velocidad gracias a que empuja materia (gases) por su tobera. Entonces, por el principio de acción y reacción, la nave se acelera en el sentido contrario en que expulsó los gases. Por supuesto, la materia que se expulsa, el combustible, es limitada y tarde o temprano se acaba, por eso es necesario optimizar los impulsos que se aplican al motor y realizar sólo los estrictamente necesarios.

Este motor principal es el que acelera a la nave para desplazarse largas distancias, y como no hay aire que la frene, es suficiente realizar un impulso durante un tiempo hasta alcanzar una alta velocidad (de miles de km/h) y luego mantenerse en esa velocidad sin seguir acelerando, con el propulsor ya apagado. Lo que deberá realizar cuando esté llegando al punto de destino es usar el mismo propulsor y aplicar exactamente la misma fuerza

durante el mismo tiempo que al comienzo hasta anular la velocidad que ganó anteriormente.

La utilización del mismo propulsor para acelerar en un sentido o en otro se logra girando la nave entera para ubicar al motor en la dirección en que se desea aplicar la fuerza. La nave en el vacío no tiene la limitación de un avión o un cohete en la atmósfera que siempre tiene que mantener una orientación con respecto a la dirección de movimiento para no perder sustentación. Como ya se explicó, en el vacío una vez que la nave se mueve a una determinada velocidad y dirección se mantendrá a esa velocidad y dirección hasta que se le aplique otra fuerza, y puede, mientras tanto, rotar en cualquier ángulo que no influirá en su trayectoria. De hecho, la inclinación o actitud que se seleccionen para un trayecto dependerá, por ejemplo, de la posición de las ventanas para optimizar la entrada de luz o seleccionar lo que se desea visualizar o también para elegir que parte de la nave recibirá el calor desde el Sol (Figura 8).

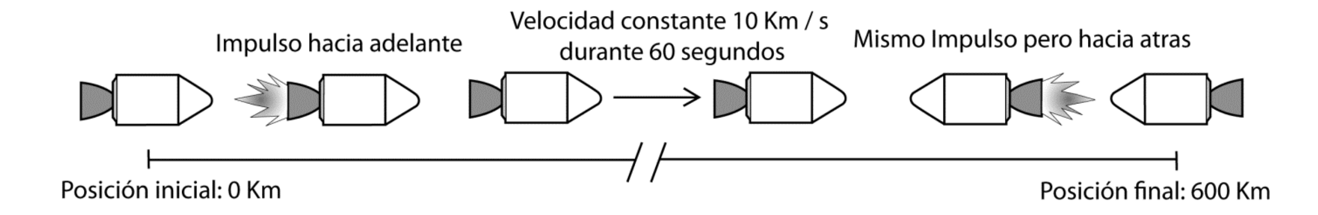



**Figura 8.** Nave espacial desplazándose 600 Km en un minuto.

Para seleccionar la orientación de la nave, o actitud, se necesitan otros propulsores que aplican fuerzas a distintas secciones de la nave y la hacen rotar sobre su centro de masa. Por supuesto que no es necesario alcanzar grandes velocidades con estos propulsores porque se usan sólo para realizar giros, no para viajar grandes distancias, por eso son pequeños y están distribuidos por distintos puntos de su superficie. Cuando se quiere girar en determinado ángulo, se aplica una fuerza similar en dos propulsores en lados opuestos de la nave para hacerla ganar un poco de velocidad angular. Cuando la nave se encuentra en el ángulo adecuado, se disparan propulsores opuestos a los anteriores y con una fuerza similar hacen que la velocidad angular vuelva a 0, y así la nave queda apuntando en la posición deseada (Figura 9).

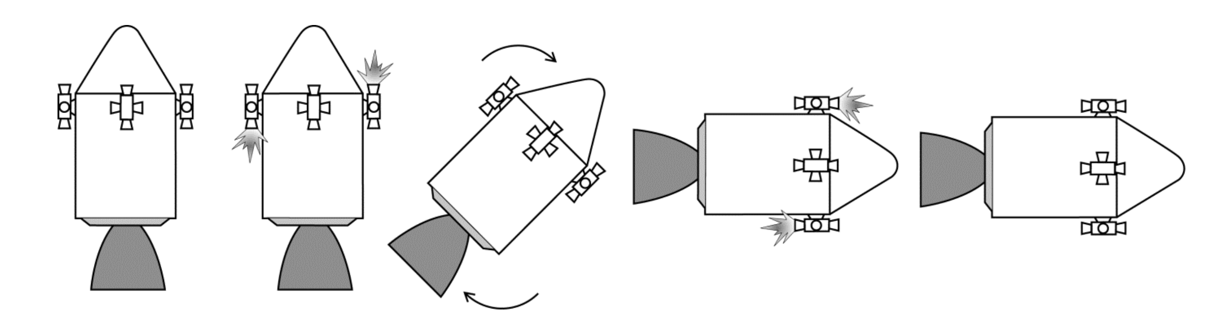


**Figura 9.** Propulsores de Módulo de Comando y ejemplo de giro.

De esta forma la nave puede apuntar en cualquier dirección y luego acelerar con el motor principal. Estas maniobras de rotación también son limitadas por el combustible, pero no consumen tanto e incluso se tienen propulsores redundantes para que la nave pueda realizar cualquier rotación en cualquier momento a pesar de que alguno de ellos no funcione correctamente.

## Órbitas

Se podría pensar que otra característica del vuelo en el espacio es que allí no hay gravedad, pero esto es incorrecto: el campo gravitatorio de un cuerpo, como por ejemplo la Tierra, se extiende hasta el infinito. Así es que la Luna es atraída por el campo gravitatorio terrestre a 380.000 km de distancia y la Tierra es atraída por el campo gravitatorio solar a 150 millones de kilómetros. Si lanzamos un satélite hasta una altura de 400 km y lo soltamos, el satélite caerá sobre la Tierra y se destruirá. La razón por la que hay satélites a 400 km y no se estrellan sobre la Tierra, como tampoco lo hace la Luna, es porque están orbitando a la Tierra. Se están aproximando constantemente sobre el planeta, pero como a la vez tienen una velocidad perpendicular a la dirección de la gravedad lo suficientemente alta, mientras caen se desplazan lateralmente y nunca chocan con la superficie. Muchas veces se utiliza el término *orbitar* como sinónimo de estar en el espacio, pero orbitar es algo específico, es un estado de equilibrio con la gravedad de un cuerpo.

Para orbitar un cuerpo a una determinada altura se necesita alcanzar una determinada velocidad. Mientras más cerca se desee orbitar, donde el campo gravitatorio es más intenso, se requerirá una mayor velocidad perpendicular a la gravedad.

Así es que en la Tierra existen distintos tipos de órbita, por ejemplo a 400 km se realiza una vuelta completa en 90 minutos a una velocidad de 8 km/s, pero a 36.000 km la vuelta completa se da en 24 horas a 3 km/s. Por eso, esta última es la órbita de los satélites geoestacionarios, que se mueven a la misma velocidad con que rota la Tierra y siempre permanecen en un mismo punto sobre la superficie. No es posible tener un satélite a 36.000 km de altura que de una órbita completa en 90 minutos, ni uno a 400 km que complete la órbita en 24 horas porque la velocidad depende de la distancia al cuerpo que se orbita.

El reto para lograr poner en órbita un satélite es principalmente conseguir alcanzar la altura adecuada y a una velocidad lo suficientemente alta para entrar en órbita: unos 8 km/s a 400 km de altura en el caso del Sputnik. En las misiones Apolo ocurre lo contrario al querer entrar en órbita alrededor de la Luna, porque la velocidad con la que llega la nave es superior a la velocidad orbital necesaria. Por eso, debe dispararse su propulsor principal en sentido contrario al movimiento para bajar la velocidad hasta alcanzar la velocidad orbital.

A causa de las limitaciones del uso de combustible en el viaje espacial, es que se recurre todo el tiempo a las órbitas para la actividad espacial. Si se pudiera sería más práctico tener un satélite de comunicaciones a 400 km de altura fijo sobre un punto a elección, pero la única forma de mantenerse estable en el espacio sin consumir combustible es entrar en órbita y atenerse a sus reglas.

### *Maniobras orbitales*

La trayectoria que sigue un cuerpo en órbita, tal como la describió Kepler en el siglo XVI, es una elipse, siendo un círculo un tipo especial de elipse en que cada punto está a la misma distancia del centro. Una órbita excéntrica es una órbita no circular y tiene como propiedad que en un punto en particular se está más cerca del cuerpo alrededor del cual se orbita (este punto se llama periápside, perigeo en el caso de orbitar la Tierra y perilunio para la Luna) y en el extremo opuesto se está lo más lejos posible (llamado apoápside o apogeo en la Tierra y apolunio en la Luna).

Las órbitas ocurren en un plano dentro del espacio tridimensional (una elipse es una figura de 2 dimensiones), por eso, además de la distancia y forma de la órbita, también puede variar el plano en el que ocurren. Por ejemplo el plano orbital sobre el que la Tierra gira alrededor del Sol es distinto al plano en que la Luna gira sobre la Tierra. Si estuvieran en el mismo plano ocurrirían eclipses todos los meses porque se alinearían muy seguidos al Sol, la Tierra y la Luna (Figura 10).

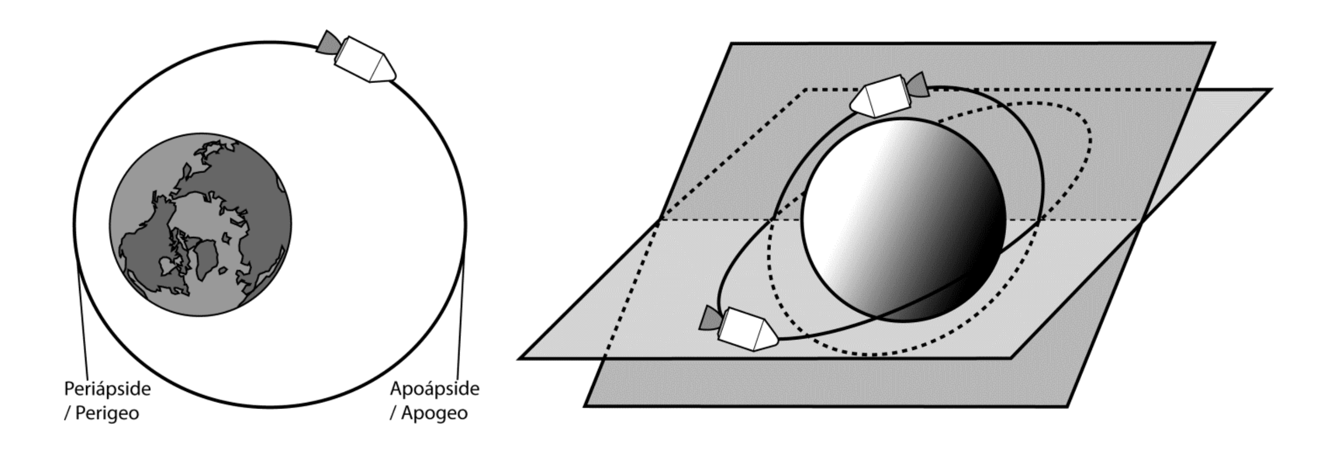



**Figura 10.** A la izquierda órbita elíptica alrededor de la Tierra. A la derecha un ejemplo de dos naves orbitando un planeta en órbitas similares pero en dos planos orbitales distintos.

En el viaje espacial es necesario cambiar de plano orbital y es una maniobra que consume combustible porque la nave debe ganar velocidad en esa orientación en particular. El plano orbital con el que se parte desde Tierra depende del lugar desde donde se despegó: si se partiera desde un punto en el Ecuador el plano cortaría en dos a la Tierra por esa misma línea. El Ecuador es justamente el lugar de partida ideal porque se aprovecha la velocidad de rotación de la Tierra, que es mayor allí, para ganar velocidad. Por ese motivo, los países que lanzan cohetes al espacio eligen un lugar de despegue que cumpla varios requisitos de seguridad pero además que esté lo más próximo al Ecuador para poder partir en un plano con la menor inclinación posible a este plano que corta en dos a la Tierra. En el caso de Estados Unidos este lugar es Cabo Cañaveral. La Estación Espacial Internacional orbita la

Tierra en un plano que es intermedio de los despegues de Estados Unidos en Cabo Cañaveral y de Rusia en Baikonur (en territorio de Kazajistán), de esta forma los principales socios pueden alcanzar esa inclinación con un mínimo consumo de combustible. En las misiones a la Luna es necesario cambiar de plano orbital porque si no todas las misiones estarían limitadas a posarse sólo sobre algún punto en la franja de la superficie lunar por donde pasaría este único plano.

Otra maniobra necesaria en el viaje espacial es el cambio de órbita. Por ejemplo, pasar de una órbita circular a 400 km de distancia de la Tierra a una a 2.000 km también circular (Figura 11). Por supuesto que siempre existe la posibilidad de ir con la nave hasta un punto de la órbita nueva y ahí acelerar hasta la velocidad necesaria, pero esto no es óptimo en cuanto a uso de energía. Una propiedad de las órbitas es que si en un punto de la trayectoria se aplica un impulso en el sentido de movimiento o en contra de él, o sea un cambio de la velocidad, se modificará la trayectoria completa. Por ejemplo, si se aumenta la velocidad en un punto de la órbita circular, a partir de ese punto la trayectoria se irá abriendo con respecto a la trayectoria anterior hasta que en el punto opuesto se llega a la máxima distancia al planeta (apeápside). A partir de allí, la nave empezará a perder altura hasta llegar al mismo punto donde realizó el cambio de velocidad a exactamente la misma altura y la órbita se repetirá así por siempre (Figura 12a). Para convertir esta órbita elíptica en una órbita circular con una altura igual a la máxima de la elíptica (apeápside), será necesario al llegar a este punto de mayor altura realizar otro cambio de velocidad para modificar el resto de la trayectoria (ver Figura 12b). De esta forma con dos impulsos se puede pasar de una órbita circular a otra también circular pero a distinta altura.

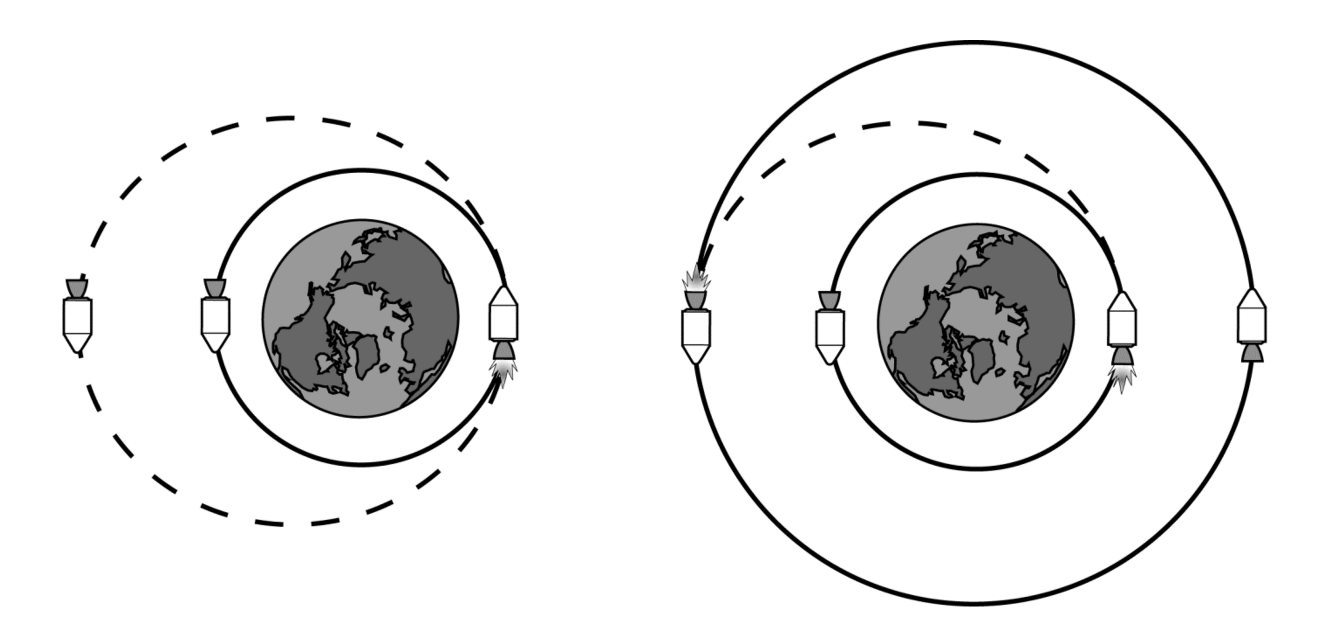


**Figura 11.** A la izquierda ejemplo de cambio de órbita circular a órbita elíptica. En la derecha cambio de orbita circular a órbita circular de mayor altura por medio de una órbita temporal.

También se recurre a cambios de órbita similares para que una nave que orbita la Tierra realice la reentrada a la atmósfera, en este caso aplicando un impulso para ir a una órbita más baja que pasa por atmósfera terrestre. Existen distintos tipos de maniobras que se pueden realizar con estos principios para poder hacer cambios de órbitas, por ejemplo transferencias de Hohmann o transferencias Bielípticas, cada uno con sus pros y sus contras sobre todo en duración y consumo de energía.

Un caso importante para realizar la misión a la Luna es cuando se tienen dos naves orbitando un cuerpo celeste y se necesita que se encuentren las dos aplicando maniobras de cambio de órbitas para poder coincidir las dos en un mismo punto en el mismo instante. Pongamos como ejemplo el caso de dos naves que orbitan la Tierra, ambas en un mismo plano y misma órbita a 400 km de altura con una velocidad de 8 km/s dando una vuelta completa en 90 minutos, pero la segunda nave pasa por un punto 30 minutos después que la otra (Figura 12.a). Una forma que se podría pensar para que la primera nave se encuentre con la siguiente es que reduzca su velocidad hasta 0 y espere 30 minutos a que la otra nave pase por ese mismo punto (Figura 12.b). Lo que ocurriría en la realidad es que la nave, al eliminar su velocidad perpendicular a la gravedad, caería directo hacia la Tierra y 30 minutos después cuando la otra nave pase por el punto donde se detuvo la primera, esta ya esté destruida (Figura 12.c). Casos similares ocurren con cualquier cambio de velocidad que una nave haga para encontrarse con la otra en la misma órbita, al modificar la velocidad se altera la órbita que sigue la nave y así pasado el tiempo no se encontrarían porque estarían en trayectorias distintas. Una forma matemáticamente simple de que logren encontrarse sería que la primera nave apunte hacia atrás y aplique una fuerza que haga que vaya a la misma velocidad (8 km/s) pero en sentido contrario y 15 minutos después, antes de estrellarse de frente con la otra nave, realizar el mismo cambio de velocidad. La contra, por supuesto, es que implica dos cambios de velocidad gigantescos (de 8 km/s a –8 km/s y de –8 km/s a 8 km/s otra vez) y un consumo de combustible enorme.

La solución eficiente pero complicada es que una o ambas naves realicen cambios de órbitas. Si la nave que está adelantada pasa a una órbita de mayor altura que realiza una rotación completa a la Tierra en un tiempo mayor, achicará la distancia con la nave que viene detrás (Figura 12.d). Por supuesto el encuentro se debe realizar estando las dos naves en la misma órbita y con velocidades iguales (Figura 12.e) sino esta diferencia las destruiría al tocarse. Proyectar estos cambios de las trayectorias para realizar el encuentro de dos naves es una tarea que requiere muchos cálculos, hay muchas variables que hay que tener en cuenta y la idea es encontrar la solución óptima para el problema, tanto para resolverlo usando la menor cantidad posible de combustible como a la vez que se pueda ejecutar en un lapso de tiempo aceptable, no luego de dar decenas de órbitas y esperar varios días.

Las órbitas elípticas son cerradas, la trayectoria es un circuito que se repite una y otra vez, pero aparte de este tipo de órbitas existen además las abiertas, en donde sólo se pasa por un punto una sola vez y luego el objeto se aleja hasta el infinito. Estas trayectorias ocurren cuando se sobrepasa la velocidad de escape del planeta, una velocidad que cuando es superada hace que la gravedad del planeta ya no pueda volver a capturar al cuerpo. Por

ejemplo si arrojamos una piedra hacia arriba, la piedra se mueve en esa dirección pero constantemente pierde velocidad hasta que deja de alejarse y comienza a caer, y si se arroja con mayor velocidad tardará más en volver pero igual caerá. En cambio si la velocidad con la que se la arroja supera la velocidad de escape del planeta (11,2 km/s en el caso de la Tierra) por más que la gravedad la atrae y constantemente le reduce la velocidad que lleva hacia arriba, de la misma forma que en los otros casos, la velocidad es tanta que a medida que va aumentando la distancia y la atracción gravitatoria es menos intensa, no habrá manera aun transcurrido un tiempo infinito para que la piedra deje de tener una velocidad que se aleja y pase a tener una velocidad en dirección a la Tierra.

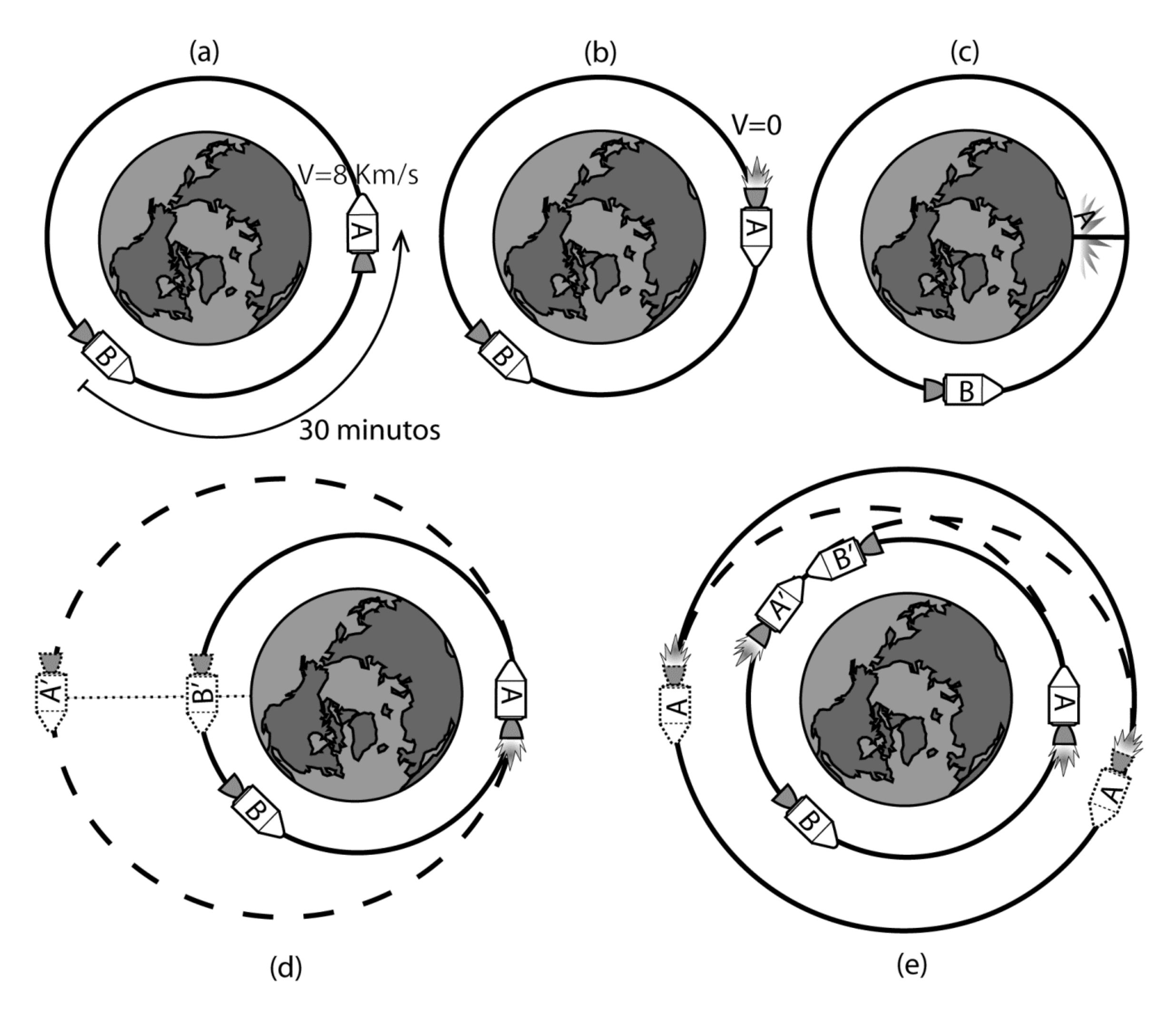



**Figura 12.** (a) Dos naves en la misma órbita con 30 minutos de separación.(b) La nave A frena en un punto para esperar a la B.(c) Al haber perdido la velocidad la nave A dejó de orbitar y se estrelló en la Tierra.(d) Sincronización de velocidad angular por cambio de órbita de una nave, ambas naves se encuentran sobre el mismo punto sobre la Tierra pero a distintas alturas.(e) Encuentro de las dos naves con dos cambios de órbita, A primero pasa a una órbita más amplia y luego vuelve a descender sobre la misma órbita.

En el trayecto hacia la Luna se utiliza una órbita similar a una abierta, aunque no hace falta alcanzar la velocidad de escape porque la meta no es simplemente abandonar la Tierra,

sino llegar a la región del espacio donde el campo gravitatorio lunar es más intenso que el terrestre y se comience a caer hacia la Luna en lugar de hacia la Tierra.

Todas las maniobras orbitales consisten en realizar un cambio de velocidad en un punto en particular de la órbita. Para poder llevar a cabo estas maniobras es necesario conocer bien la posición y velocidad que lleva la nave, su masa y el campo gravitatorio donde está orbitando y luego, en base a esta información, se puede calcular cómo debería ser el impulso, o cambio de velocidad, para modificar la órbita actual. Se necesita tener poder de cálculo para obtener esta información y luego es necesario ejecutar la maniobra en el lugar adecuado apuntando en la dirección correcta.

***Matemática de las órbitas***

La matemática que describe el comportamiento de cuerpos en órbitas tiene su origen en la ley de gravitación universal formulada por Isaac Newton en 1687. Esta ley establece la fuerza entre dos cuerpos a causa de la mutua interacción gravitatoria (Figura 13).

$$F = G\frac{m_1 m_2}{d^2}$$

**Figura 13.** Ley de gravitación universal, la fuerza de gravedad entre dos cuerpos es proporcional a sus masas e inversamente proporcional al cuadrado de la distancia entre ellos.

Todas las características de las órbitas, ya sean las formas que describen, las velocidades requeridas y distintos tipos de maniobras surgen de aplicar esta fórmula matemática. La física necesaria para estudiar las trayectorias espaciales pertenece al siglo XVII, pero se necesitaron muchos desarrollos matemáticos para poder aplicar estas fórmulas a casos concretos. La ley se aplica fácilmente a dos objetos puntuales y la aplicación a planetas ideales (completamente esféricos) fue demostrada por el propio Newton por medio del desarrollo del análisis matemático, pero los cálculos se complican mucho cuando se consideran más de dos cuerpos interactuando entre sí, por ejemplo la Luna orbitando alrededor de la Tierra y estos dos juntos orbitando al Sol. Si se toma el sistema Luna-Tierra por separado y luego se lo analiza en su órbita alrededor del Sol como si fuera un solo cuerpo, se dejan de lado detalles sutiles pero que en el largo plazo se hacen notorios, como los efectos de la gravedad del Sol sobre la órbita de la Luna. Este problema se conoce como el problema de los tres cuerpos y llevó varios siglos de estudio en que se buscó una fórmula matemática que representara el estado de los tres cuerpos en cualquier momento, hasta que en 1887 Henri Poincaré demostró que no existe una formula general para este problema: no es un sistema periódico, nunca repite su estado en el tiempo, y en su análisis sentó las bases para la teoría del caos que estudia sistemas en los que pequeñas variables hacen que en el largo plazo no se puedan predecir el estado que tomará el sistema completo.

Con la aparición de las computadoras se hizo posible realizar simulaciones del sistema solar por medio de la ejecución de miles o millones de operaciones matemáticas aplicando las leyes de la física. Gracias a estos cálculos se pudieron obtener soluciones numéricas aproximadas a los distintos desafíos aparecidos con la exploración espacial. Se hizo posible estudiar distintas trayectorias posibles para los viajes por el sistema solar y buscar las formas más óptimas para moverse dentro de los campos gravitatorios.

Pocos años después del programa Apolo, las mejoras en los desarrollos matemáticos permitieron utilizar la gravedad de los distintos planetas para empujar sondas a lugares lejanos del sistema Solar. Las dos sondas Voyager [11] realizaron larguísimos viajes hacia los planetas exteriores aprovechando una inusual alineación de estos e inclusive, la Voyager II ya ingresó en el espacio interestelar. El cálculo de donde estaría cada planeta en cada momento se realizó con tanta precisión que la Voyager II, partiendo desde la Tierra, se dirigió primero a Júpiter, donde la gravedad y velocidad de este planeta la empujaron hacia Saturno utilizando una técnica conocida como impulso gravitatorio. Saturno la empujó hacia Urano que a su vez le dio un impulso para ir en dirección a Neptuno y toda esta trayectoria la realizó aproximándose lo suficiente a cada planeta para fotografiarlo. Los encuentros con los planetas ocurrieron con años de diferencia entre ellos y la trayectoria a través de los campos gravitatorios se llevó a cabo con tanta exactitud que la sonda pasó por cada uno de los planetas en el momento indicado, tomando en cada caso la velocidad que este llevaba y la acumulación de todos estos impulsos la mantuvo en la senda prefijada incluso después de 12 años, cuando se topó con el último de los planetas que debía visitar.

## NAVEGACIÓN ESPACIAL

La tarea principal para la que fue pensada la computadora de las naves Apolo, es para la navegación en la ruta hacia la Luna: para saber con precisión en cada momento hacia qué dirección apunta, en qué punto del espacio se encuentra, a qué velocidad se está moviendo, cuál es la magnitud de la aceleración que está modificando su velocidad y con toda esa información poder realizar las maniobras adecuadas en el lugar y el momento preciso.

### Sistemas de coordenadas

La nave debe saber en cada momento su posición utilizando 3 coordenadas $(x, y, z)$ para identificar un punto en el espacio tridimensional. El punto a considerar como el origen a partir de donde se mide y como se orientan los tres ejes del sistema de coordenadas va a depender del uso que se piense darle. Por ejemplo si se toma el eje $z$ como la recta que pasa por Cabo Cañaveral, donde despega el cohete, puede ser muy útil durante el momento de ascenso para ver el avance del cohete con respeto al punto de partida, pero en el trayecto hacia la Luna, donde la nave ya no gira junto con la Tierra, tomar las coordenadas en un sistema que rota de forma independiente no es práctico. En cambio, resultaría mucho más

útil tomar un eje que una la Tierra y la Luna y medir la posición en base a él. Lo mismo pasa cuando la nave orbita la Luna, un sistema de coordenadas que esté centrado en un punto de referencia de la Luna y gire con ella resultaría adecuado para saber la trayectoria que sigue la nave sobre su superficie.

Utilizar distintos sistemas de coordenadas no es complicado si se sabe cómo convertir una posición expresada en un sistema a otro, y la computadora, por medio de operaciones vectoriales, realiza esta operación de forma inmediata, por eso es conveniente usar en cada momento el sistema que mejor represente la trayectoria que se está siguiendo. Se necesita un sistema de referencia para orientar los sistemas de coordenadas que se utilicen en la navegación. En la Tierra siempre sabemos dónde es arriba y dónde abajo porque es un eje que es obvio. El resto del trabajo está en obtener una referencia para saber dónde está, por ejemplo, el Norte y a partir de ahí, con esas dos líneas o vectores, se puede deducir un sistema de coordenadas de tres dimensiones para la superficie y con el cual nos podemos orientar en cualquier dirección. En el espacio es más complejo por la no existencia de un eje arriba-abajo ni de puntos cardinales que dependen del eje de rotación la Tierra. Se necesita definir un sistema de coordenadas de tres dimensiones independiente de la Tierra y las estrellas fijas son una excelente opción para tomar como referencia de dirección. En la Tierra la posición de las estrellas depende de la hora del día, por la rotación sobre su eje, pero en el espacio se puede tomar a las estrellas como un sistema fijo, porque en la práctica es posible considerar que sus posiciones no varían según nuestra ubicación de tan lejos que están.

***Sistema de navegación inercial***

Un sistema de navegación inercial lo que permite es calcular la dirección, aceleración, velocidad y posición de un vehículo (una nave espacial en este caso) procesando la información de los instrumentos que es obtenida del propio movimiento del vehículo sin tener ninguna referencia externa. En un vehículo sin comunicaciones de ningún tipo ni ventanas o cámaras para observar el mundo exterior, el sistema inercial le debería alcanzar para conocer su ubicación y velocidad, siempre y cuando parta desde una posición conocida a una velocidad conocida, y el resto del tiempo se actualiza esta información con lo leído de los instrumentos.

Esta información se calcula en una computadora utilizando como instrumentos giroscopios y acelerómetros y el conjunto de todo este sistema se conoce como unidad de medición inercial (IMU en la sigla en inglés).

En la determinación de la posición de la nave debe tomarse en cuenta la velocidad a la que se está moviendo porque esta modifica todo el tiempo su posición. La computadora del sistema de navegación sabe a la velocidad en que se mueve la nave y constantemente está sumando esta velocidad a la posición. Por ejemplo si sabe que se tiene una velocidad de 10 km/s, cada segundo que pasa debe sumarle esos 10 km a las coordenadas que mantiene de la nave, o si lo hace 10 veces por segundo le sumará en cada caso 1 km.

Para conocer la velocidad a la que se mueve la nave, el sistema inercial lo que hace es aplicarle a la velocidad la aceleración leída en cada instante además de la aceleración del campo gravitatorio donde está ubicada.

**Los giroscopios**

Un giroscopio es un instrumento que gira a gran velocidad sobre un eje *y* por su alto momento de inercia mantiene su orientación en ese eje. Si el giroscopio está montado en un armazón que le permite orientarse en una posición independientemente de los movimientos que tenga el lugar donde se encuentra, entonces este instrumento permanecerá apuntando siempre en la misma dirección. Por ejemplo si en el Ecuador se deja en el suelo un giroscopio a las 12 del mediodía apuntando hacia abajo (Figura 14) a las 6 de la tarde, como la Tierra giró 90 grados sobre su eje, el giroscopio apuntará paralelo al piso y a las 12 de la noche el extremo que miraba hacia abajo estará dirigido hacia el cielo. Esto es así porque el instrumento se mantiene girando siempre sobre su propio eje independientemente de los movimientos de la Tierra debajo de él.

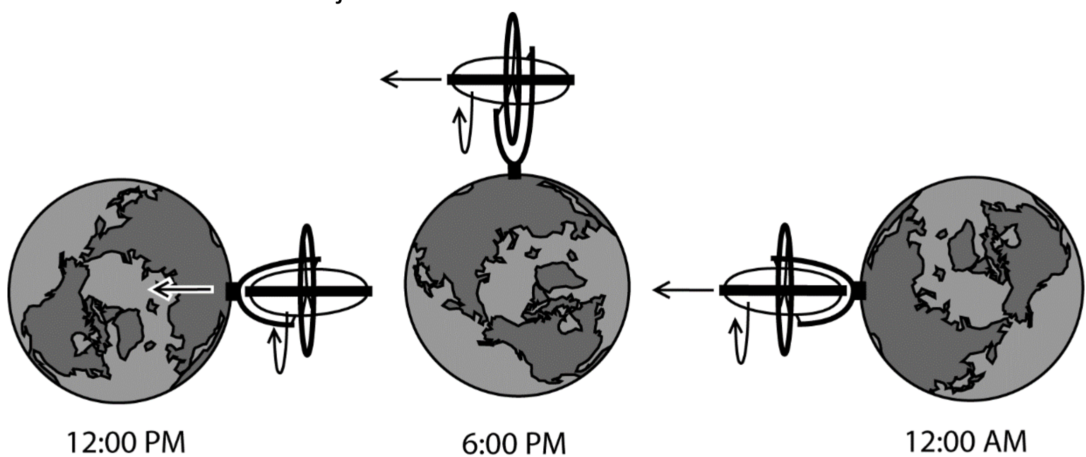



**Figura 14.** 12 P.M.: El giroscopio apunta hacia el suelo. 6 P.M.: El giroscopio apunta paralelo al piso. 12 A.M.: sigue apuntando hacia el mismo lado pero con 180 grados de diferencia con respecto a la Tierra.

Un instrumento de este tipo mantiene la dirección en un eje *y* si se fija a otro objeto va a hacer que este se mantenga también estable en ese eje y hará falta ejercer una fuerza mayor para cambiar la orientación del objeto en ese eje en comparación a los otros (Figura 15).

Un uso que se le puede dar a un giroscopio es para indicarle a un motor eléctrico cómo mantener una estructura siempre en la misma orientación en un eje: el giroscopio está en una plataforma pero en un armazón con movimiento independiente en un eje, y cuando la plataforma y el giroscopio están alineados este instrumento no envía ninguna señal, pero cuando se desalinea en determinada dirección sí envía una señal y cuando lo hace en la dirección opuesta envía otra señal. Cuando recibe una de estas señales el motor realiza un giro en la dirección opuesta para compensar el giro que hizo desalinear la plataforma con respecto al giroscopio. El motor va a girar siempre que el giroscopio no esté perfectamente

alineado con la plataforma para oponerse al movimiento que produjo la desalineación y buscar el equilibrio.

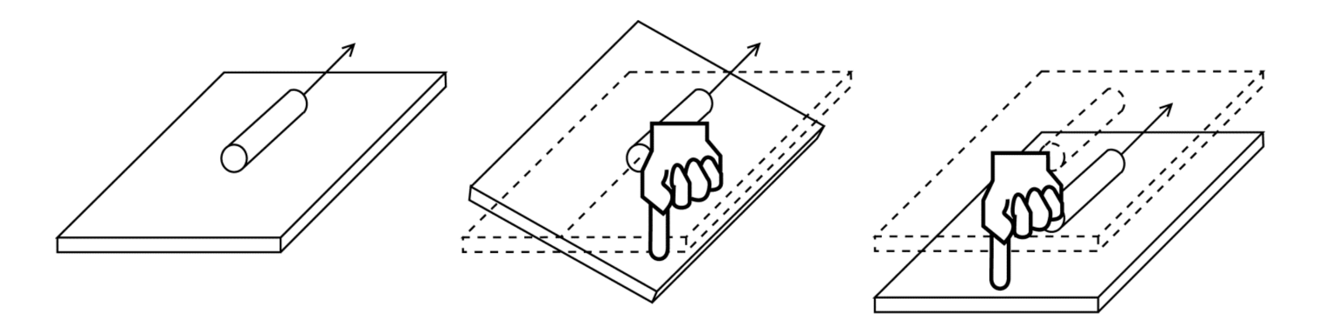


**Figura 15.** Al a izquierda una plataforma flotando en el agua con un giroscopio en el centro. En el medio se aplica una fuerza a un costado del eje estable y la plataforma gira sobre este eje. A la derecha se presiona sobre el mismo eje y la plataforma entera mantiene la orientación, lo que hace es hundirse toda por igual.

Si se hace lo mismo pero con tres giroscopios y tres motores se puede mantener una plataforma siempre orientada en el espacio tridimensional sin importar los movimientos externos a la plataforma. Si se pone una estructura así dentro de una nave espacial, esta superficie quedará apuntando siempre en la misma dirección, por ejemplo con respecto a las estrellas, y se puede usar como sistema de referencia absoluto, como el centro del sistema de navegación inercial que se mantendrá orientada sin importar los movimientos de la nave que lo contiene. Midiendo los ángulos de cada eje de esta plataforma estable se conoce la orientación que lleva la nave con respecto al sistema de coordenadas del sistema inercial. Si además de los tres giroscopios se le agregan a la plataforma estable tres acelerómetros, cada uno apuntando en un eje de coordenadas, se pueden medir las aceleraciones que ocurren en cada eje con respecto al sistema de coordenadas en uso, sin importar la orientación de la nave (Figura 16).

En la actualidad los giroscopios mejoraron mucho tanto en tamaño y peso y además de los giroscopios rotatorios existen instrumentos electrónicos sin partes movibles que cumplen la misma función y se pueden incluir dentro de un circuito integrado. Por su bajo costo se hicieron muy accesibles y se utilizan mucho para vehículos controlados electrónicamente, como los drones, pero también se usan en dispositivos portátiles como teléfonos celulares y tablets, y en algunos controles para juegos.

**Acelerómetro**

Un acelerómetro es simplemente un dispositivo que mide la aceleración que ocurre sobre sí mismo. Es muy similar a una balanza de resorte, consiste en un peso conocido que al medir su desplazamiento obtiene la aceleración que está actuando sobre él. El acelerómetro nos dice la magnitud y dirección de la aceleración sobre un eje, siendo la aceleración la tasa de cambio de la velocidad o la derivada de la velocidad en el tiempo.

Pongamos como ejemplo un auto que se mueve en el eje X y tiene un acelerómetro orientado en ese eje (Figura 17).

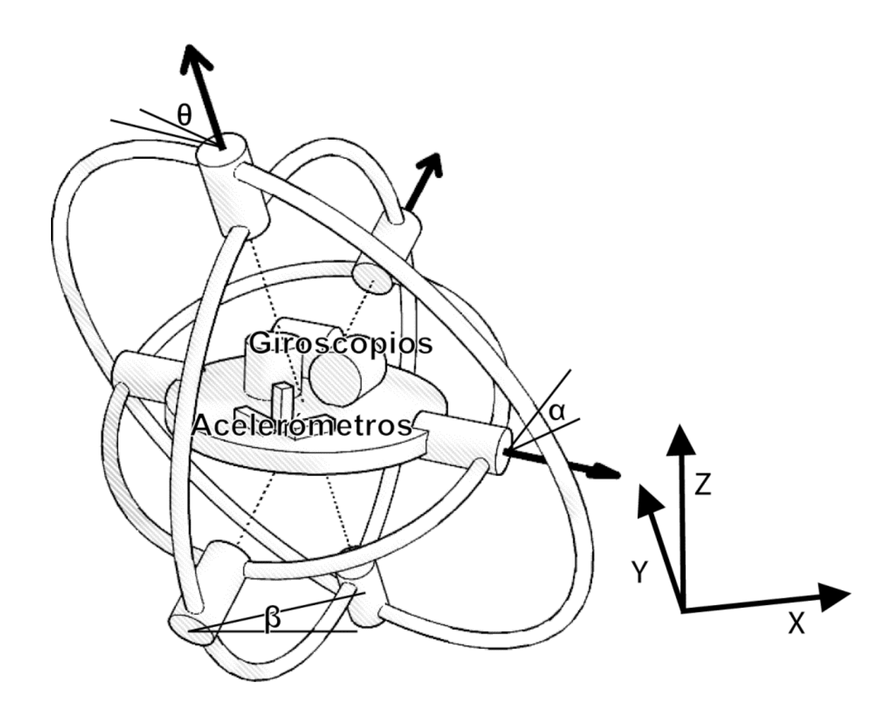



**Figura 16.** Una plataforma estabilizada con tres giroscopios. La parte estable está en el centro sostenida por 3 brazos movibles que le dan total libertad de movimientos con respecto al vehículo en donde se encuentra. Midiendo los ángulos α, β y θ en los ejes de los brazos se puede obtener la orientación de la nave con respecto a la plataforma estable. Dentro de la plataforma estable hay tres acelerómetros, uno para cada eje de coordenadas, para medir las aceleraciones que ocurren en todas las direcciones.

En el segundo 0 el auto está quieto, la velocidad es 0 (en nuestro sistema de referencia), la posición *x* es 0 (también en nuestras coordenadas) y no hay aceleración.

Desde los 10 segundos y hasta los 20 se presiona el acelerador y se produce una aceleración positiva. La velocidad aumenta (linealmente) y la posición aumenta en forma de parábola.

Luego del segundo 20 se sigue con el acelerador presionado pero la fuerza del motor se compensa con las fuerzas que resisten su avance (resistencia del aire, de las ruedas, del motor, etc.) lo que deja como resultante una aceleración nula. La velocidad queda constante desde la última vez que hubo aceleración, no vuelve a 0.

En el segundo 50 se presiona el pedal del freno y se produce una aceleración negativa. Durante 8 segundos se reduce la velocidad, pero al terminar el frenado la velocidad permanece constante y todavía mayor a 0, aunque es más lenta de lo que era antes del frenado.

En todos los casos la posición en el eje *x* siempre fue creciente en el tiempo aunque con una velocidad variable, que en algunos casos fue positiva y en otros negativa.

Para este ejemplo utilizamos un muestreo de tiempo de muy poca definición, cada 5 segundos, lo cual puede no ser práctico porque cualquier aceleración menor a ese tiempo

podría ser ignorada. La velocidad de muestreo debe ser la suficiente para incorporar las aceleraciones más breves que puede detectar el sensor (Tabla 1).

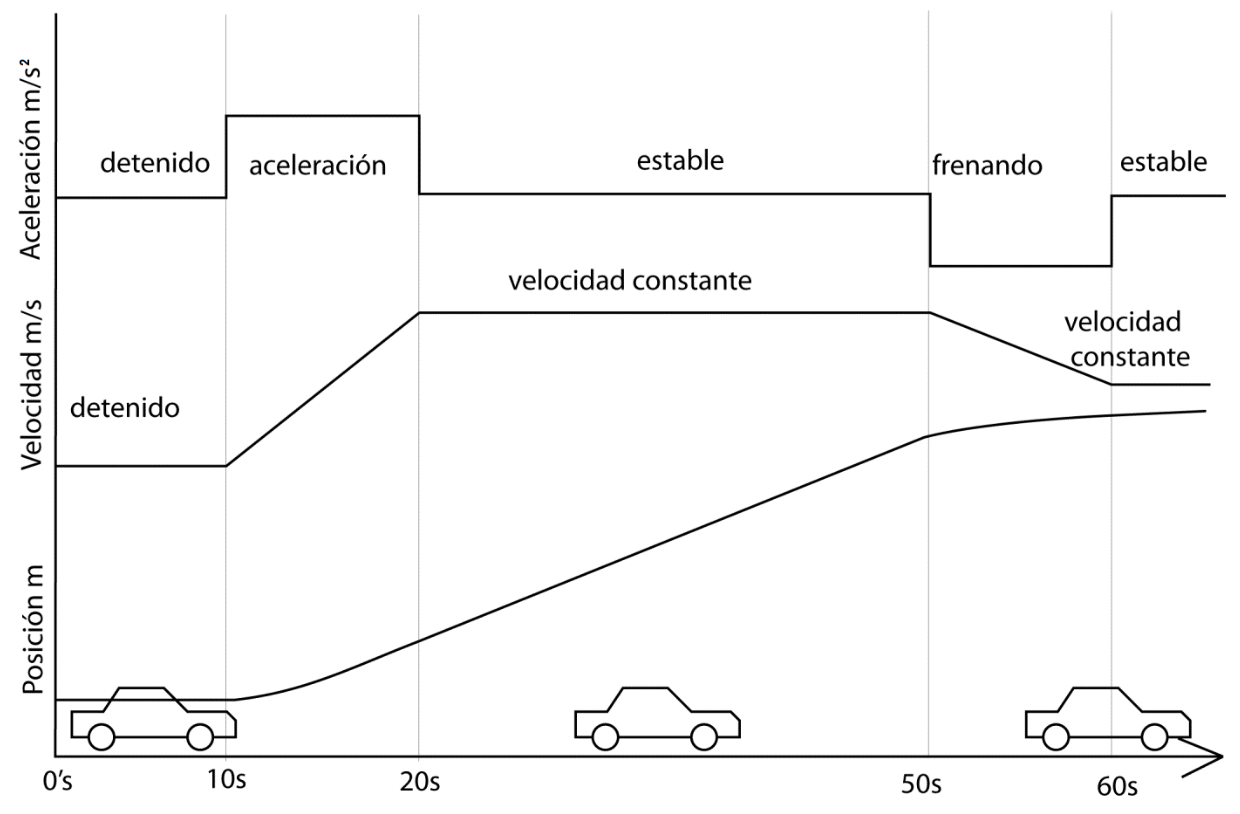



**Figura 17.** Movimiento simplificado de un auto en el eje X.

A la medición obtenida del acelerómetro el sistema inercial la suma a la velocidad del eje en el que apunta el acelerómetro. Luego el sistema utiliza esa velocidad para ajustar periódicamente la posición del sistema.

Este tipo de uso del acelerómetro para obtener la velocidad y posición de un objeto por medio de una computadora (microcontrolador pequeño normalmente) es cómo se utiliza por ejemplo en los controles de los juegos tipo Nintendo Wii o de una tablet, y en automóviles en el sensor de un freno ABS para regular la aplicación del freno, o el sensor de que activa el airbag si la desaceleración fue excesiva (una magnitud tal que implica que la parte delantera del auto ya empezó a deformarse).

### *Acelerómetros en la superficie terrestre*

En el ejemplo anterior se mostró como trabaja un acelerómetro sobre un eje y sería lógico pensar que para aplicar lo mismo a un sistema de 3 dimensiones alcanzaría con usar 3 acelerómetros, cada uno apuntando en uno de los ejes de coordenadas, pero todos realizando los mismos cálculos, aunque en realidad la situación se complica cuando hay que lidiar con la fuerza de gravedad, que en el ejemplo del auto se ignoró por tratarse de un eje horizontal.

| Tiempo (s) | Aceleración ($m/s^2$) | Aceleración acumulada últimos 5 s | Velocidad (km/s) | Velocidad promedio últimos 5 s | Posición (m) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 |
| 10 | 2 | 0 | 0 | 0 | 0 |
| 15 | 2 | 10 | 10 | 5 | 5 |
| 20 | 0 | 10 | 20 | 10 | 15 |
| 25 | 0 | 0 | 20 | 20 | 35 |
| 30 | 0 | 0 | 20 | 20 | 55 |
| 35 | 0 | 0 | 20 | 20 | 75 |
| 40 | 0 | 0 | 20 | 20 | 95 |
| 45 | 0 | 0 | 20 | 20 | 115 |
| 50 | -1.6 | 0 | 20 | 20 | 135 |
| 55 | -1.6 | -8 | 12 | 16 | 151 |
| 60 | 0 | -8 | 4 | 8 | 159 |

**Tabla 1.** Parámetros deducidos desde la lectura del acelerómetro. Tomando como referencia el tiempo y la aceleración medida en cada momento se calcula la velocidad y con esta la posición.

Una propiedad del acelerómetro es que no puede medir la aceleración causada por la fuerza de gravedad porque esta fuerza actúa, no sólo sobre la parte externa del acelerómetro o el vehículo que lo contiene, sino que además ejerce la misma fuerza sobre todos los componentes internos del instrumento a causa de que estos también poseen masa. Como la fuerza de gravedad afecta a todos los elementos del acelerómetro y no hay diferencias de fuerzas para comparar no hay posibilidad de obtener ninguna medición.

Pongamos un ejemplo para ver cómo se mide la aceleración en un acelerómetro alineado con el eje vertical (Figura 18). El primer caso es cuando el acelerómetro está suelto en el aire y cae (Figura 18.a). Esto se denomina caída libre, nada lo sostiene y la aceleración hacia abajo es la que produce la gravedad en la superficie terrestre, 9,8 $m/s^2$ (1 G) pero esto lo podemos ver si lo medimos externamente realizando la observación desde un punto de referencia en reposo: desde el acelerómetro lo que se mide es 0, porque la misma aceleración que se produce en la parte externa del acelerómetro se produce adentro y no hay nada que medir. En el segundo caso, se ubica al instrumento en un montacargas y permanece quieto y sostenido (Figura 18.b): la aceleración medida desde afuera es 0 porque no hay movimiento, pero el acelerómetro mide 9,8 $m/s^2$ hacia arriba que es la fuerza de la plataforma resistiéndose a la aceleración. Los componentes internos del acelerómetro son atraídos por la gravedad pero no pueden moverse por la oposición del montacargas y esta tensión es la se mide. Luego el montacargas acelera hacia abajo (Figura 18.c) porque el cable hace menos fuerza para oponerse a la gravedad. Desde afuera se observa una aceleración de 4 $m/s^2$ hacia abajo pero el acelerómetro mide 5,8 $m/s^2$ h hacia arriba, debido a que hay una fuerza menor oponiéndose a la gravedad. El otro caso es cuando el cable tira más fuerte que

el campo gravitatorio (Figura 18.d): una medición externa dará 5 m/s$^2$ en dirección hacia arriba y el acelerómetro obtendrá también una aceleración hacia arriba pero de magnitud 14,8 m/s$^2$. El último caso (Figura 18.e) es cuando la plataforma con el acelerómetro se desalinea y el instrumento queda apuntando en una dirección distinta a la de la gravedad. La aceleración medida estando quieta la plataforma es menor a 9,8 m/s$^2$, serán 6,7 m/s$^2$ si el ángulo de inclinación es de 45 grados.

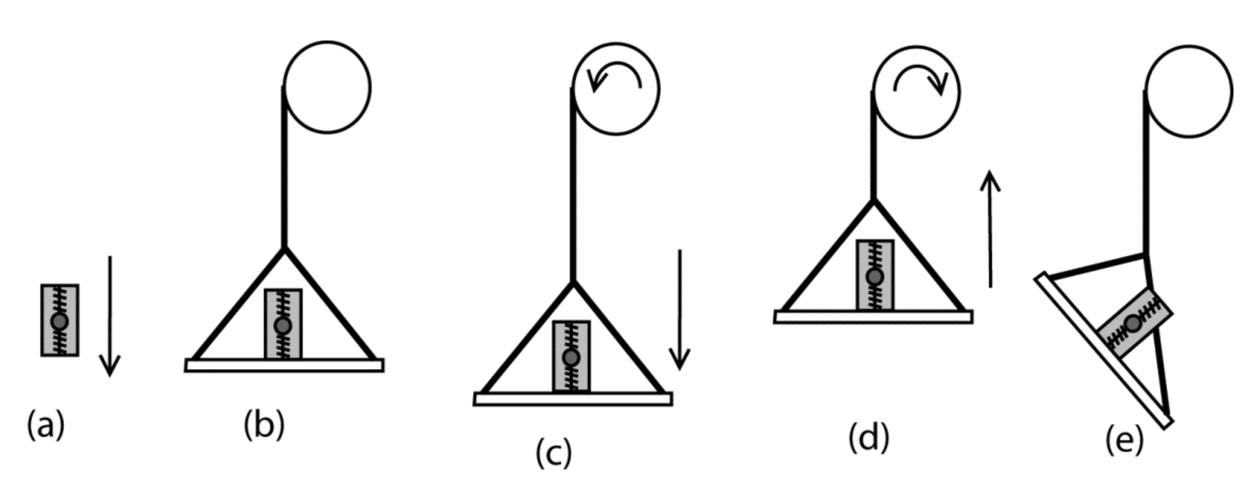



**Figura 18.** Acelerómetro en la Tierra en caída libre y en un montacargas.

En los primeros 4 casos la solución para medir la aceleración correcta es restarle siempre los 9,8 m/s$^2$ de la aceleración de la gravedad que el acelerómetro no puede medir: en todos los ejemplos el instrumento mide 9,8 m/s$^2$ de más a causa de esto. En el último caso, con el acelerómetro desalineado, las soluciones para obtener información válida de sus mediciones son, o bien alinearlo o bien conocer el ángulo de inclinación y restarle la aceleración por la gravedad que le corresponde por este ángulo, y en ambos casos tener un giroscopio ayuda porque sirve para alinear al acelerómetro o para conocer el ángulo con respeto a la gravedad.

La implementación de un sistema inercial se complica por esta propiedad de los acelerómetros, sobre todo en la lógica para controlarlos que debe conocer la alineación con el campo gravitatorio y aplicar continuamente esta fuerza a las mediciones para obtener resultados válidos. No existe otra forma de medir un campo gravitatorio además de, como ya se mencionó, realizar observaciones sobre un punto de referencia que no se esté acelerando a causa de la gravedad (por ejemplo mirar el piso), pero un instrumento que por sí solo detecte la presencia y magnitud de un campo gravitatorio no es posible, de hecho el *Principio de Equivalencia* formulado en la *Teoría Especial de la Relatividad* por Albert Einstein afirma que no hay forma de distinguir un campo gravitatorio de un sistema de referencia que está siendo acelerado. Por ejemplo no podríamos darnos cuenta si estamos en una habitación en la Tierra o en un ascensor en una estación espacial que acelera 9,8 m/s$^2$ en una dirección.

La característica que tienen los acelerómetros para medir si están siendo sostenidos o en caída libre la utilizan muchos equipos portátiles como notebooks y celulares para

detectar que se están cayendo. Cada equipo incorpora al menos un acelerómetro de 3 ejes muy pequeño (está hecho con silicio en un circuito integrado). Constantemente mide las aceleraciones que se le aplican y es utilizado en programas y juegos para realizar detección de gestos, movimientos, etc. Si el equipo se deja caer, el acelerómetro idealmente mediría 0 G de aceleración en cada eje (en la práctica lee un poco más porque el aire algo lo sostiene y amortigua la aceleración). Al detectar este estado sin aceleración, o sea que nadie lo sostiene (caída libre), el equipo entra en un modo de emergencia que desactiva lo antes posible la alimentación de los periféricos y la posición del cabezal del disco rígido para minimizar los destrozos producidos en el impacto que se aproxima.

***Acelerómetros en distintos puntos del campo gravitatorio terrestre***

Según la ley de gravitación universal, la fuerza gravitatoria varía con la distancia entre los cuerpos, lo que hace que su magnitud no sea constante para todo el espacio alrededor del planeta (Figura 19). Se suele utilizar la fórmula de campo gravitatorio que está basada en la de la gravedad pero con la masa de un solo cuerpo (la Tierra en este caso) y da como resultado en vez de una fuerza, la aceleración que sufriría un objeto si estuviera ubicado en un determinado punto alrededor del planeta.

$$\vec{E} = -G \frac{M}{|\vec{r}|^2} \hat{r}$$

**Figura 19.** Formula de campo gravitatorio de la Tierra. Indica para cada punto (vector r) la aceleración hacia el centro del planeta que tendrá un cuerpo ubicado en ese punto. Es una simplificación porque no se tienen en cuenta variaciones del campo por la forma irregular de la Tierra.

Si se trabaja siempre sobre la superficie terrestre se pueden ignorar la fórmula de campo y suponer que la aceleración es siempre 9,8 $m/s^2$ (1 G) como se hace en el ejemplo del montacargas. Pero para el sistema de navegación de una nave que se va a mover por el espacio, es necesario que sepa en todo momento a que distancia está de la Tierra, y utilice esta distancia para calcular la magnitud y dirección del campo gravitatorio en ese punto.

Es vital que el sistema de navegación conozca con la mayor precisión posible su posición con respecto a la Tierra para poder calcular la aceleración que está actuando sobre la nave, y a la vez el modelo matemático de la gravedad debe ser muy fiel a la realidad para que la aceleración calculada sea la correcta, porque una desviación de este dato producirá una velocidad incorrecta y luego la posición del sistema se actualizará en base a esta velocidad errónea y con una posición incorrecta se obtendrá una aceleración todavía más alejada a la realidad que la anterior, y así con el tiempo la desviación con la posición real se hará cada vez más grande.

### *Acelerómetros en órbita*

Como se vio anteriormente, cuando se orbita se está en caída libre, por eso un acelerómetro no medirá ninguna aceleración estando en órbita y toda la información dependerá de que el modelo matemático de la gravedad del sistema inercial se ajuste lo suficiente a la realidad.

Por ejemplo una nave que está orbitando la Tierra a 400 km de altura no obtendrá ninguna información de los acelerómetros, siempre leerá 0, y toda la información utilizada para calcular la posición deberá partir de la aplicación del modelo matemático por medio de la computadora (como ya se explicó, la aceleración de la gravedad es calculada por la computadora en base a su posición), de la aceleración se calcula la velocidad, con la velocidad se actualiza la posición y luego con este dato actualizado se obtiene la nueva aceleración y así continuamente. Esta serie de operaciones requiere de un modelo matemático lo más próximo que sea posible a la realidad porque es la única referencia interna para poder calcular la posición y velocidad.

Antes se mencionó el caso de las notebooks con el sistema de acelerómetros para detectar caídas. Cabe destacar que un dispositivo de estos presentaría problemas si se pone en órbita, porque justamente estará en caída libre, lo que en la Tierra tarde o temprano termina mal, pero estando órbita es lo normal. Esta opción se debe desactivar en el equipo si no no va a funcionar en órbita.

### *Acelerómetro entre los campos gravitatorios terrestre y lunar*

En la práctica nunca hay un solo campo gravitatorio aislado del resto, sino que en cada punto del espacio se suman los distintos campos de los planetas y cuerpos celestes que nos rodean. En la Tierra la gravedad de la Luna y el Sol producen efectos notables como las mareas, a pesar de la gran distancia a la que se encuentran.

Se puede ignorar el efecto de la gravedad de la Luna para aplicaciones terrestres por su pequeña magnitud, pero si se habla de un viaje de la Tierra a la Luna se tienen que incorporar los dos campos gravitatorios al modelo matemático a utilizar (Figura 20).

$$\vec{E} = G\left( -\frac{M_T}{|\vec{r}|^2}\hat{r} + \frac{M_L}{|\vec{v}-\vec{r}|^2}\frac{\vec{v}-\vec{r}}{|\vec{v}-\vec{r}|} \right)$$

**Figura 20.** Campo gravitatorio de la Tierra sumado al de la Luna, con origen de coordenadas en el centro de la Tierra, donde $M_T$ es la masa de la Tierra y $M_L$ de la Luna, r es el vector con la posición donde se evalúa el campo, con el centro de coordenadas en la Tierra en este caso y v es la posición de la Luna en el mismo sistema de coordenadas.

El sistema de navegación debe saber en qué punto en el espacio está con respecto a la Tierra y a la Luna para saber cómo lo afecta la gravedad de cada uno de estos cuerpos y es

importante conocer la hora con precisión para obtener la posición de la Luna con respecto a la Tierra, porque la Luna se mueve y hay que tenerlo en cuenta debido a que su campo gravitatorio se desplaza con ella.

Al sumar los campos de los dos cuerpos se obtiene uno menos simétrico que el de un solo planeta. En algunos puntos la gravedad neta sumada apunta hacia la Tierra y en otros puntos apunta hacia la Luna (Figura 21).

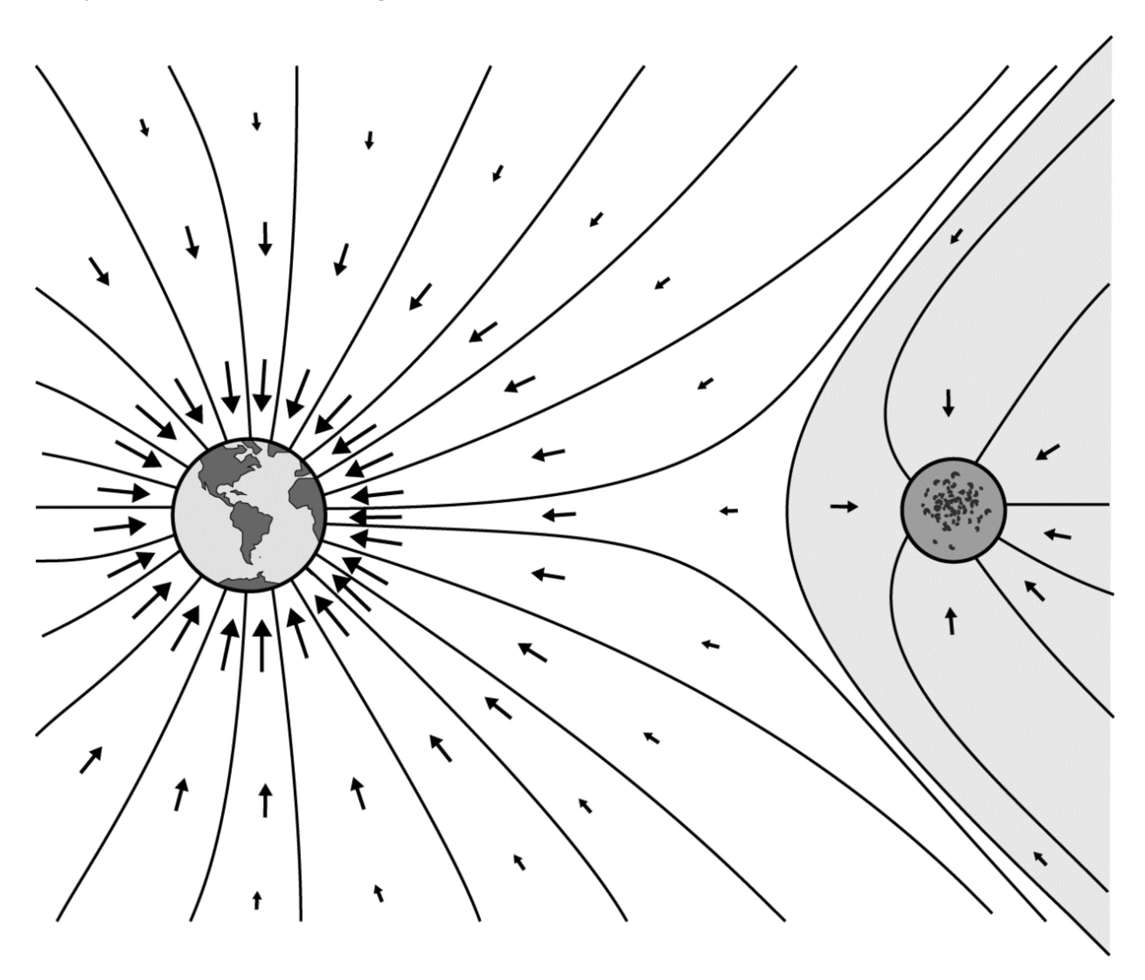


**Figura 21.** Campo gravitatorio entre la Tierra y la Luna. Las flechas indican intensidad y dirección del campo gravitatorio. El área sombreada es donde la gravedad neta apunta hacia la Luna y es mucho más chica que el área donde se apunta hacia la Tierra porque esta última tiene un campo mucho más intenso. La ilustración no está a escala.

## Calibración del sistema inercial durante el vuelo

Un sistema de navegación inercial se puede hacer muy preciso con instrumentos de buena calidad y modelos matemáticos que se aproximen mucho a los campos gravitatorios reales, pero aun así existen errores que se van acumulando a lo largo de un viaje largo. Los giroscopios se van corriendo hacia un lado por imperfecciones de su construcción (movimiento de precesión), el cálculo de la posición en base a las aceleraciones de la nave se va desfasando por pequeños defectos del acelerómetro que en miles de kilómetros se acumulan y también el reloj de referencia no es infinitamente preciso. El modelo matemático de los campos gravitatorios es una aproximación: hay variaciones en el campo

que se notan más estando cerca de la Tierra o la Luna, porque la distribución de la masa no es homogénea y pueden haber desviaciones grandes, sobre todo en la Luna donde el campo es muy variable y no está tan estudiado.

Por todo esto es necesario realizar calibraciones del sistema de navegación. No se puede tomar como referencia absoluta al sistema inercial, que es muy útil en distancias relativamente cortas, pero la acumulación de errores, como vimos, lo hace poco confiable en plazos largos. Una forma de calibrar el sistema inercial es por medio de la observación del cielo, de la posición de las estrellas y cuerpos del sistema solar.

### *Calibración por medio de las estrellas*

La navegación por medio de la observación del cielo se viene realizando desde hace miles de años, incluso mucho antes de la invención de la brújula. Las técnicas de navegación fueron progresando con la creación de instrumentos más precisos como el telescopio, la publicación de mapas celestiales cada vez más exactos y el desarrollo de relojes de mayor confiabilidad. Con los años se fue haciendo posible utilizar rutas marítimas más directas en lugar de navegar siempre junto a las costas gracias a que se podía obtener más información de la observación del cielo, y esta tecnología es un factor muy importante en la historia humana, sobre todo en la creación de las grandes redes de comercio mundial.

Los objetos a observar en el cielo nocturno se dividen en dos grupos: los pertenecientes a nuestro sistema solar, como los planetas y los cometas y luego están las estrellas lejanas, fuera de nuestro sistema solar. Las distancias relativas de todo lo que se encuentra dentro del sistema solar son mucho menores que las demás estrellas. La distancia al Sol de los demás planetas, incluso los más alejados, es comparable con la distancia de la Tierra al Sol. Por eso a medida que la Tierra gira alrededor del Sol, dando una vuelta completa en un año, desde cada posición de su órbita se ven a los demás planetas en posiciones distintas según la distancia en que esté cada uno. Incluso Neptuno, que está 30 veces más lejos del Sol que la Tierra y tarda 164 años en dar una órbita completa (por lo que parecería casi quieto en el cielo), observándolo desde distintos puntos de la órbita terrestre se puede obtener la distancia a la que se encuentra comparando las mediciones.

No ocurre lo mismo con las estrellas lejanas, su distancia a la Tierra está en una escala totalmente distinta a las usadas dentro de nuestro sistema solar y dan la sensación de estar fijas cada una en una posición en el cielo. La estrella más cercana a nuestro Sol es Próxima Centauri y está a 4,2 años luz, 15000 veces más alejada del Sol que la Tierra. Su ángulo de paralaje, o sea el ángulo en que varía la posición de la estrella en el cielo mirándola desde distintas posiciones, es el mayor que se puede medir de una estrella. Realizando observaciones con una separación de 6 meses (con 300 millones de km de distancia entre las dos posiciones de la Tierra) se miden 1,5 segundos de arco de ángulo de paralaje, siendo un segundo de arco 1/3600 de grado. Este ángulo es tan pequeño que se necesita un telescopio muy poderoso para poder medirlo.

Por esta razón a las estrellas lejanas se las puede considerar como fijas y es posible utilizarlas como sistema de referencia para orientarse en el espacio.

### *Calibración del ángulo de la nave con respecto las estrellas fijas*

Para calibrar la orientación del sistema de navegación se utilizan las estrellas fijas, necesitando identificar al menos dos estrellas conocidas para poder orientar el vector de dirección. Las estrellas se las identifica por los patrones que forman en el cielo, por ejemplo por su posición dentro de una constelación o con respecto a ella.

En la superficie terrestre, ya sea en el mar, en tierra firme o en el aire, es posible orientarse identificando una sola estrella conocida, porque siempre se tiene como referencia un vector en el eje vertical, simplemente en la dirección de la gravedad. Las estrellas que se suelen usar para orientarse sin necesidad de realizar cálculos vectoriales para convertir sistemas de coordenadas son, en el hemisferio sur, la constelación de la Cruz del Sur y, en el norte, la estrella Polaris (comúnmente llamada Estrella del Norte o Polar) de la constelación Osa Menor, porque están ubicadas muy cerca del eje de rotación de la Tierra y sabiendo cuál es el Sur o el Norte y conociendo dónde es arriba y abajo es trivial obtener un sistema de coordenadas. Además, aquí en la Tierra, a causa de su rotación, la posición de las estrellas en el cielo depende de la hora del día y el día del año, salvo para estas estrellas que están en el mismo eje de rotación que siempre mantienen su posición en el cielo.

En cambio el sistema de referencia que se usa en el espacio es independiente de la hora en la Tierra, es un sistema absoluto con respecto al espacio. Con identificar dos estrellas, que desde el punto de observación forman un plano en el espacio, se puede deducir un tercer eje perpendicular a las dos. Al contar con una computadora no es necesario que las estrellas que se toman como referencia estén apuntando en uno de los tres ejes del sistema vectorial usado por el sistema inercial, sino que pueden apuntar a cualquier lugar. La computadora conoce las coordenadas de cada estrella usando el mismo sistema de coordenadas para todas y luego por medio de operaciones matemáticas con vectores es fácilmente convertible un sistema de coordenadas en otro (Figura 22).

El sistema inercial, una vez que obtuvo un sistema de referencia de dirección, puede reorientar los giroscopios para que apunten en la dirección correcta y los acelerómetros estén correctamente alineados.

### *Calibración de la posición*

Una vez calibrada la orientación se necesita calibrar la posición en que se encuentra la nave en el espacio.

Desde la superficie terrestre se mide la posición de una estrella conocida, tomando su ángulo con respecto al horizonte con un sextante, y sabiendo con la mayor exactitud disponible el día y la hora es posible calcular las coordenadas en la Tierra, gracias a que se conoce los movimientos de rotación del planeta y de ahí el movimiento de las estrellas en el

cielo terrestre a lo largo del año y el día: una determinada medición a una determinada hora sólo puede ser realizada desde un único punto sobre la Tierra.

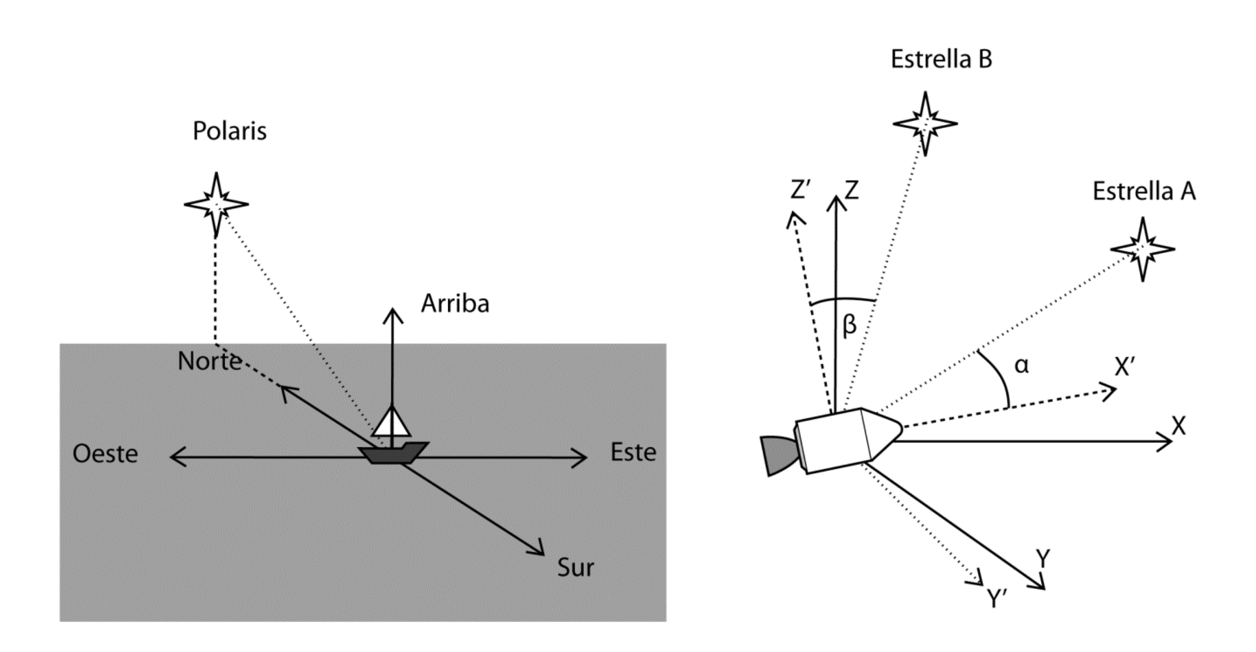



**Figura 22.** Orientación por medio de la observación de las estrellas, en el mar (a la izquierda) y en el espacio (derecha).

En el espacio, como ya se vio, midiendo los ángulos de las estrellas no es posible obtener la posición en que nos encontramos, porque, incluso con un telescopio de grandes dimensiones y midiendo las estrellas más cercanas, los ángulos que se pueden medir son muy pequeños, con las mejores mediciones sólo se podría calibrar la posición en millones de kilómetros, lo que no es muy útil.

Para obtener la posición lo más simple es medir un cuerpo cercano porque se puede observar con mayor detalle. En un viaje entre la Tierra y la Luna cualquiera de estos dos cuerpos nos sirve como referencia. Utilizando un sextante, se miden los ángulos que forman un punto en la Tierra de coordenadas conocidas con una estrella también conocida. Con este ángulo, teniendo en cuenta el día y hora en que fue realizada la medición, existe un solo punto en el espacio tridimensional desde donde se pudo realizar esta medición en ese instante dado. La computadora, utilizando su reloj y su conocimiento de la posición de la Tierra según la fecha y la hora, realiza inmediatamente el cálculo de la posición desde donde se realizó esta observación (Figura 23).

Un punto de referencia de la Tierra que se puede tomar fácil porque es visible a cualquier hora y nunca está cubierto con nubes es el mismo horizonte de la Tierra, si no es necesario tener una lista amplia de puntos identificables para poder usar en cualquier circunstancia. Esto es básicamente lo mismo que se hace para medir la posición desde la

Tierra: se compara la posición de las estrellas con respecto al horizonte de la Tierra y se toma como dato la hora.

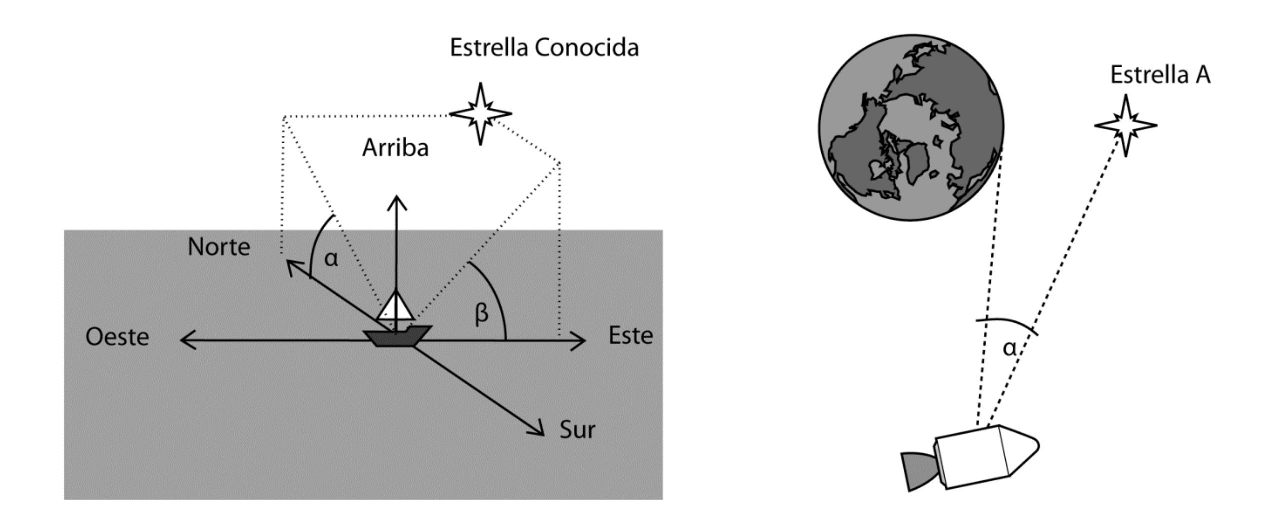



**Figura 23.** A la izquierda determinación de la posición en la Tierra midiendo los ángulos de una estrella. A la derecha medición del ángulo de una estrella con el horizonte terrestre desde la nave para determinar su posición sabiendo la fecha y hora.

## Implementación del sistema de navegación

Para crear el sistema de navegación se toman todos los instrumentos que forman la unidad inercial y la computadora, que será el cerebro del sistema, y se deben integrar de forma tal que desde la computadora se puedan obtener todas las mediciones posibles. El siguiente paso, conociendo las leyes que rigen la navegación en el espacio, consiste en escribir un software que aplique todo ese conocimiento. El programa tiene que tener toda la lógica del sistema de navegación inercial, que por medio de la lectura de los giroscopios y acelerómetros y al modelo matemático de los campos gravitatorios donde se encuentra, y tomando como referencia el reloj de la computadora, aplique todos los algoritmos para calcular la aceleración, de esta la velocidad y luego la posición. También debe proveer la opción para que los tripulantes puedan calibrar la orientación y la posición por medio de las lecturas que realizan los astronautas en el sextante y tiene que facilitar este trabajo incluyendo una tabla de estrellas de coordenadas conocidas para realizar los cálculos.

La información del sistema inercial se debe utilizar para el control de la nave, para, por ejemplo, disparar un propulsor cuando se encuentre en la dirección adecuada y lograr la trayectoria buscada. El software necesita tener en cuenta detalles más finos, como, por ejemplo, la estimación del peso del combustible en cada momento en base al combustible que ya se consumió, y el dato este lo debe utilizar para calcular de forma más precisa el impulso que necesitará la nave para obtener determinado cambio de velocidad. El sistema es un sistema de lazo cerrado, esto significa que el software cuando envía una acción, por ejemplo dar ignición a un propulsor para producir una aceleración, a la vez obtiene feedback, en este caso realiza mediciones con el sistema inercial y por medio de estas

mediciones puede ver cómo avanza la operación que inició. Midiendo la aceleración del impulso del propulsor puede calcular cuando detener el propulsor al alcanzar la velocidad requerida. A la vez el software conoce la mecánica de los propulsores y sabe que si al iniciar la ignición luego de un determinado tiempo no se midieron aceleraciones entonces hay un problema en algún sistema que controla al propulsor y debe alertar a la tripulación para que verifiquen el estado de todos los sistemas relacionados (Figura 24).

Los datos principales de la trayectoria que debe seguir la nave fueron precalculados en Tierra con anticipación y así fue como se planificó la cantidad de combustible a utilizar y el peso del equipamiento a llevar. El sistema de navegación de la nave buscará ajustarse a esta trayectoria (o trayectorias en caso de tener que pasar a una ruta alternativa en una situación de emergencia) lo más fielmente que sea posible.

Este tipo de vehículos es ideal para ser controlado por software a diferencia de, por ejemplo, un auto en una ciudad. Conducir en la calle requiere de la percepción humana para evaluar obstáculos. Recién ahora se está empezando a avanzar en el diseño de autos que se manejan por computadora pero utilizando equipos muy poderosos y una cantidad exagerada de cámaras, sonares, radares, receptores satelitales y sensores de distinto tipo apuntando a distintos lados, todos integrados para obtener la misma información que un humano procesa con facilidad con dos ojos. Para controlar al auto se necesita tener la atención al 100% porque existen muchos tipos de obstáculos a evitar que pueden aparecer de forma sorpresiva, como peatones cruzando de forma incorrecta, una rotura en la calle o una mala maniobra de otro vehículo más adelante. Los aviones están en un punto intermedio porque utilizan rutas que están garantizadas para no encontrarse con otros aviones pero se debe estar controlando el vuelo constantemente sabiendo que en cada momento puede ocurrir un evento climático que haga variar las condiciones del vuelo. Además, los motores están todo el tiempo empujando y hay que controlarlos permanentemente, a diferencia del vuelo espacial donde se usa casi todo el tiempo la inercia y son contados los momentos en que se disparan los propulsores.

Un software que navegue por el espacio exterior no necesitará estar revisando el estado de las cosas muchas veces por segundo porque es un ambiente muy predecible. El principal factor que guía la trayectoria es el campo gravitatorio y se ajusta con mucha precisión al modelo matemático. Aun con una computadora poco poderosa se pueden evaluar los efectos de la gravedad cada varios segundos y aplicar los cambios en velocidad y posición recién allí. Este ambiente que es tan difícil de comprender intuitivamente por una persona es ideal para una computadora con un programa adaptado a él.

En el espacio no es probable encontrarse con obstáculos en el camino, si bien hay micrometeoritos, asteroides y otros objetos en el sistema solar, están dispersos en un volumen muy grande y la posibilidad de colisión es muy baja, y aun si se intentara buscar obstáculos y evadirlos, haría falta un sistema de detección avanzado para poder ver con la suficiente anticipación cualquier objeto a las velocidades altísimas con que se mueven los cuerpos en el vacío, y serían necesarios avanzados sistemas de radar que consumen mucha energía, e incluso en el caso de que se detectara un objeto en la trayectoria de la nave, el

impulso para cambiar la trayectoria con rapidez sería enorme. Está de más decir que en el viaje a la Luna de las misiones Apolo y en el sistema de control de la nave no había ninguna alternativa de esquivar obstáculos: se limitaba a seguir la trayectoria lo más fielmente posible y en el muy improbable caso de que hubiera algo justo en la trayectoria de la nave no habría ninguna posibilidad de salvar a la tripulación.

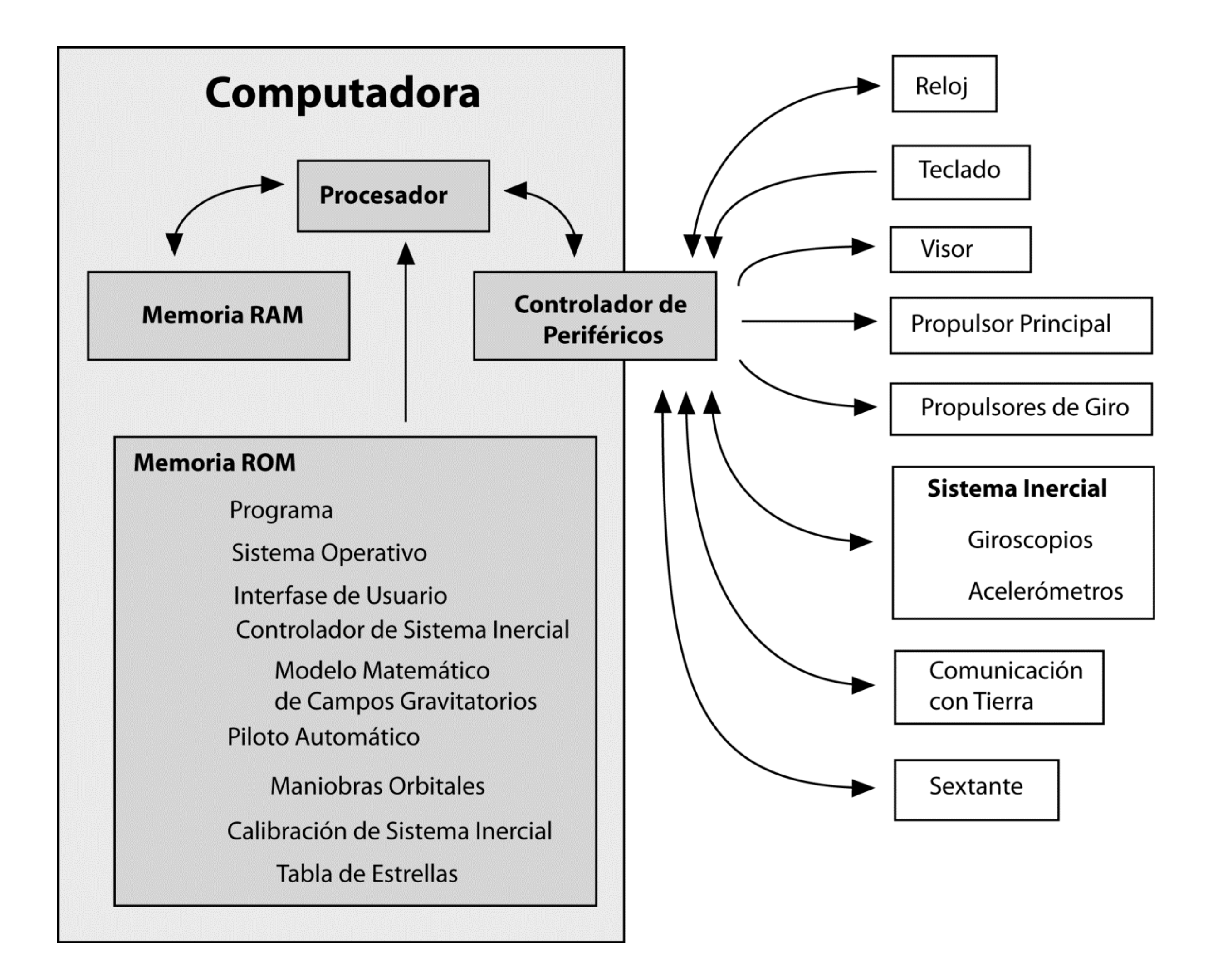



**Figura 24.** Esquema de componentes principales del sistema de navegación de las naves Apolo, con la computadora y los periféricos que esta controla y de donde obtiene información. Las flechas muestran en qué dirección se puede mover la información. Los casos de doble sentido son los que permiten leerse y también modificarse, por ejemplo de la plataforma del sistema inercial se pueden leer los ángulos pero además es posible modificarlos al realizar la calibración.

## LA TAREA DEL MIT

El MIT tenía en sus manos la tarea de realizar el sistema de navegación que usarían las naves Apolo, desde la computadora hasta el sistema inercial, los instrumentos ópticos para calibrar al sistema inercial y todo el software que debía dar vida al sistema. Una simplificación en el proyecto que se decidió casi desde el principio, fue que, como el viaje a

la Luna se realizaba con dos naves independientes entre sí (Módulo de Comando y Módulo Lunar) y cada una necesitaba una computadora, se optó por diseñar una sola computadora y dotar de una a cada nave, de esta forma se unificaba el diseño del equipo y el entrenamiento de los astronautas que utilizarían la misma interfaz en cualquiera de sus puestos de comando. Las dos computadoras serían idénticas, salvo que cada una tendría varios programas propios para las maniobras que no compartiría con la otra nave, y algunos periféricos controlados por la computadora estarían en una nave y no en otra, por ejemplo el radar para alunizaje sólo estaría en el Módulo Lunar. A pesar de ser la misma computadora, la del Módulo de Comando se llama AGC (Apollo Guidance Computer) y la del Módulo Lunar se llama LGC (Lunar module Guidance Computer) pero ambas forman en cada nave, junto con la unidad inercial, el PGNCS (Primary Guidance, Navigation and Control System: sistema primario de guiado, navegación y control).

A pesar de esta simplificación, la magnitud del proyecto encarado por el MIT era inmensa, las tareas necesarias para completarlo abarcaban muchas áreas distintas trabajando todas a la vez, y en algunos casos en el límite del conocimiento técnico de la época. Normalmente, cuando se diseña una computadora se integran componentes existentes para tener un sistema que cumpla los requisitos del producto final que se busca. Por ejemplo, en la década del 1970 y 1980 las computadoras personales que fueron apareciendo de fabricantes como Commodore, Atari, Apple, IBM o Sinclair hacían uso de microprocesadores ya disponibles en el mercado, creados por otras empresas dedicadas en exclusiva a esa tarea, como Intel, MOS, Zilog o Motorola. En 1971, con la aparición del Intel 4004, entraba en escena el microprocesador, que es básicamente una computadora entera metida dentro de un circuito integrado, pero sin las memorias ni los periféricos necesarios para funcionar. El fabricante de la computadora toma un microprocesador existente y lo dota de la cantidad y tipo de memoria que supone necesaria, diseña los periféricos que utilizará y los programas que correrá. Este fabricante no debe tomar decisiones sobre las instrucciones del procesador ni los aspectos técnicos de la electrónica interna, porque todos esos detalles ya fueron resueltos por la empresa creadora del microprocesador. También quién utiliza el microprocesador se beneficia de la documentación disponible sobre su funcionamiento, de herramientas de software dedicadas a la creación de programas para esa arquitectura y también de la experiencia previa de otros proyectos, incluso de empresas competidoras que utilizan el mismo microprocesador. La base de software y documentación existente es muy importante a la hora de diseñar un sistema y se nota en el éxito comercial de algunos sistemas en comparación a otros, sobre todo la PC, que arrastra en su diseño decisiones de la década del 1970 pero sigue siendo el estándar, e incluso en el ambiente industrial también existen microcontroladores con décadas de uso, como el 8051 o el 6800 que se siguen eligiendo por la gran base de software, documentación y experiencia existente.

El diseño de la computadora para las naves Apolo no gozaría de ninguna de estas ventajas: la misma institución que diseñaría el procesador y la tecnología para fabricarlo, también se encargaría de integrarlo en una computadora con los periféricos necesarios, a la

vez que diseñarían y escribirían el software para un procesador totalmente nuevo con el que nadie tenía experiencia de trabajo.

A pesar de trabajar en el programa Apolo con hasta un año de ventaja con otros contratistas importantes, el proyecto se fue atrasando y el presupuesto multiplicó muchas veces el valor inicial. Una de las causas que se reconocen que produjo mucha incertidumbre y complicó todas las etapas de diseño, fue la falta de especificaciones claras desde el comienzo. No estaba definido cuál era la precisión necesaria del sistema de navegación ni cuales serían todas las funciones que se debían cumplir. Con el constante agregado de requisitos mientras se estaba en la etapa de desarrollo, ya no de planificación, la computadora fue creciendo en cantidad de memoria e instrucciones, cambios que a medida que se avanza en un proyecto son más y más complicados de realizar. La versión Block I de la computadora tenía 1K de RAM y 24 K de ROM (siempre hablando de palabras de 16 bits) y la Block II sería expandida a 2K de RAM y 36 K de ROM por la creciente complejidad del software.

## CONCLUSIONES

Hasta aquí vimos de qué forma ayudaría la computadora a realizar el viaje ida y vuelta hacia la Luna con una breve descripción del funcionamiento de una nave espacial, su física y sus requerimientos de navegación y control además de presentar una introducción de cómo funciona un sistema de navegación inercial, especialmente en el espacio. También vimos el contexto de la época y los desafíos tecnológicos que implicaba el proyecto para crear esta computadora. En el próximo artículo veremos cómo funcionaba la computadora, su arquitectura y decisiones de diseño del hardware.

### BIBLIOGRAFÍA

[1] A. Chaikin (1994*). A Man on the Moon*, Penguin Books.
[2] C. Riley (2010). *Apollo 11 Owners' Workshop Manual*. Haynes Publishing
[3] D. A. Mindell (2008). *Digital Apollo*. MIT Press.
[4] F. O'Brien (2010). *The Apollo Guidance Computer Architecture and Operation.* Springer Praxis.
[5] Anónimo (2016). Misil V2. http://www.v2rocket.com/start/makeup/design.html (6 de junio de 2016).
[6] W. D. Woods (2011). *How Apollo Flew to the Moon*, Springer Praxis.
[7] N. D. C. R. Duncan Copp (dir.) (2008). *Moon Machines Episode 3. The Navigation Computer.* Science Channel.
[8] MIT Tech TV (2009). A funny thing happened on the way to the moon. Richard Battin IAP Lecture, 22 de Enero de 2009. http://techtv.mit.edu/collections/aeroastro/videos/2044-a-funny-thing-happened-on-the-way-to-the-moon (16 de mayo de 2016)
[9] R. Alonso (2010) Ramón Alonso, el argentino que llevó a la Apolo 11 a la Luna. Ariel Torres (entrevista). *La Nación*, 7 de marzo de 2010. http://www.lanacion.com.ar/1240769-ramon-alonso-el-argentino-que-llevo-a-la-apollo-11-a-la-luna (10 de mayo de 2016)
[10] R. Alonso (2010). Diseño de la computadora de vuelo del Apolo XI. Simposio Argentino de Sistemas Embebidos (SASE), 3 al 5 de marzo de 2010.
[11] Jet Propulsion Laboratory (2015). Voyager. Planetary Voyage. http://voyager.jpl.nasa.gov/science/planetary.html (29 de mayo de 2016).

# LA COMPUTADORA DE LAS MISIONES APOLO
# PARTE II: HARDWARE

Marcos Leguizamón[†]

La computadora de navegación de las naves Apolo era un proyecto muy innovador para la época y requirió mucho trabajo de investigación y diseño en distintas áreas. Como debía caber en un pequeño espacio que tenía asignado dentro de la nave era necesario reducir el tamaño de la computadora todo lo que se pudiera y el tipo de tecnología a utilizar era algo vital para poder cumplir las metas de tamaño y peso. Con este fin se recurrió a lo más avanzado en electrónica en esos años que eran los circuitos integrados, de reciente invención. En el MIT tenían como tarea estudiar esta nueva tecnología y ver cuál era la mejor forma de utilizar estos componentes en el proyecto pero manteniendo la confiabilidad requerida. La tecnología de las memorias también era algo muy importante para poder disponer de suficiente información en el sistema sin exceder el volumen disponible.

A nivel diseño se necesitaba crear una arquitectura que fuera capaz de correr eficientemente un programa complejo que todavía no había sido escrito pero cuyas características ya se estaban delineando, y, a la vez, como la tecnología electrónica no permitía todavía una gran miniaturización, había que simplificar el diseño dejando sólo lo necesario para que el tamaño de la computadora fuera el adecuado. Era necesario encontrar un punto de equilibrio entre la complejidad del hardware y la del software, porque agregarle más funcionalidad al hardware aumentaría el tamaño del equipo aunque simplificaría el desarrollo del software. Por otro lado, si se le quitaban funciones al hardware sería necesario escribir programas más complejos que realizaran por software lo que no era posible por medio de la electrónica y de esta forma se hacía más complicado el proceso de desarrollo mientras la ejecución se volvía más lenta al haber más instrucciones, elevando el riesgo de que la computadora no fuera capaz de ejecutar todo lo necesario para controlar la nave en tiempo real.

Esta computadora cumple la función de pilotear la nave en la que se encuentra, principalmente controlando los motores, tanto el principal como los pequeños propulsores que ajustan el ángulo de la nave, y todo esto lo debe realizar obteniendo información de los giroscopios y acelerómetros para saber cómo se está moviendo en el espacio. Para poder

[†] RTI Latina S.R.L., leguizamon@gmail.com

interactuar con el mundo físico la computadora necesita un sistema de entradas y salidas conectado a los distintos sistemas eléctricos de la nave para poder saber en qué estado se encuentra en cada momento y también poder controlar a la nave.

## SEMICONDUCTORES

Los transistores se inventaron en 1947 en los laboratorios Bell pero la producción de estos componentes para la industria electrónica tardó en producirse unos años más. Antes de la aparición de los transistores se utilizaban válvulas de vacío para las mismas funciones: amplificar o controlar una corriente. Las válvulas eran similares en aspecto y muchas características a las lamparitas incandescentes y consumían mucha electricidad, generaban mucho calor, necesitaban tensiones eléctricas altas, ocupaban mucho espacio y se rompían muy seguidos. Los semiconductores, tanto transistores como diodos, podían reemplazar a estas válvulas con enormes ventajas pero para eso primero había que esperar que esta tecnología madurara porque los primeros transistores, que eran de germanio y se fabricaban uno por uno, no eran muy confiables. En ese sentido, la unión de los cables con el semiconductor en su interior era muy frágil, su funcionamiento dependía demasiado de la temperatura y humedad del ambiente y, además, la velocidad que podían alcanzar era menor que la de las válvulas. Por eso se comenzaron a utilizar transistores en equipos electrónicos chicos a mediados de la década de 1950, tales como radios portátiles. Para equipos más sofisticados, como las computadoras, se siguieron usando las válvulas de vacío durante varios años más hasta fines de esa década.

Una tecnología electrónica que apareció a mitad de los ‘50 fue la de núcleos magnéticos, que produjo memorias para computadoras que se usaron durante unos 20 años más (ver en *Memoria RAM*). Aparte de esta aplicación, también se usaban estos núcleos magnéticos para realizar circuitos lógicos: compuertas lógicas AND y OR formadas alrededor de un núcleo de ferrite con varios cables con señales que según la dirección de la corriente en estos cables variaba como se magnetizaba el núcleo. Este evento disparaba el valor de salida de la compuerta. La primera versión de la computadora del MIT, la que se usaría para la sonda a Marte, utilizaba esta tecnología en lugar de transistores. La velocidad era menor a la que se podía lograr con transistores pero esta tecnología era más confiable y el consumo de energía era mucho menor. Para una computadora pensada para funcionar durante años este bajo consumo eléctrico era muy útil.

### Circuitos integrados y Ley de Moore

El circuito integrado fue inventado en 1959 por Jack Kilby de Texas Instruments y trabajando de forma independiente, por Robert Noyce de Fairchild. Básicamente, es una forma de crear un circuito electrónico completo, con transistores y conexiones, en una superficie muy pequeña formando un solo componente. En 1965, Gordon Moore, que entonces trabajaba en Fairchild y más tarde fundaría Intel junto con Noyce, publicó un

artículo con lo que luego se llamaría la *Ley de Moore*. Esta ley predecía que cada dos años la cantidad de transistores que era posible tener dentro de un circuito integrado se duplicaría. El desarrollo de la tecnología de circuitos integrados se ajusta bastante bien a esta predicción, con un constante aumento de la cantidad de transistores en los nuevos chips. Para tener una noción de este progreso, el primer microprocesador, el 4004 de Intel, en 1971 tenía unos 2300 transistores, para 1976 el Z80 de Zilog tenía 8500, luego en 1978 el 8086 de Intel incluía unos 29000 y en 1989 el 80486 ya traía 1,2 millones de transistores.

Durante la primera década de existencia de los circuitos integrados, esta tecnología no fue adoptada por la industria electrónica en general. Desde el principio se utilizaron en proyectos militares o aeroespaciales, donde existía una gran necesidad de reducir los tamaños de la electrónica sin importar el costo. Para otros tipos de productos, por ejemplo radios, no se justificaba el elevado precio que tenían estos componentes tan novedosos. El otro problema de utilizar circuitos integrados era que no estaba garantizado el suministro de componentes durante toda la vida de un proyecto: esta tecnología avanzaba tan rápido que los fabricantes de componentes abandonaban constantemente los productos viejos para sacar nuevos y si el diseño de un equipo electrónico dependía de un componente en particular entonces se debería cancelar toda la producción, generando enormes pérdidas. Para evitar problemas de este tipo, antes de iniciar la producción de un equipo se estudia que este no dependa de ningún componente con un solo proveedor no confiable en el mercado y en componentes como los transistores, con varias características que los definen, es fácil encontrar más de una empresa que produzca transistores similares. El caso de los circuitos integrados es más complicado porque tienen un diseño muy específico y es más difícil que varias empresas se pongan de acuerdo para producir piezas compatibles con el mismo diseño, por eso incluso los fabricantes de computadoras de la época, como IBM, DEC o RCA, tardaron en adoptar esta tecnología y, de hecho, recién en la época en que aparecieron los primeros circuitos integrados estaban comenzando a utilizar transistores en sus computadoras y pasarían años hasta que adoptaran el uso de chips.

Cuando fueron apareciendo versiones comerciales de estos circuitos con compuertas lógicas, en el MIT comenzaron a estudiarlos para su posible utilización, primero para el misil Polaris y luego para la NASA. La computadora, ya iniciado el programa Apolo, seguía utilizando lógica magnética pero vieron una oportunidad en el uso de semiconductores para reducir el tamaño y aumentar la velocidad aunque incrementando el consumo eléctrico. Luego de realizar pruebas con distintos chips de distintos fabricantes, seleccionaron el *Micrologic* fabricado por Fairchild y propusieron a la NASA realizar una versión del prototipo de computadora que tenían en desarrollo, aunque en lugar de usar lógica con núcleos magnéticos lo harían con estos circuitos. En 1962 la NASA aceptó, a pesar de que era una tecnología muy novedosa poco probada, muy cara y que a la vez la había rechazado en otros fabricantes de equipamiento electrónico. Este primer prototipo con circuitos integrados funcionó correctamente y a partir de ahí, en adelante sólo se trabajaría con estos componentes en los distintos modelos de la computadora.

Además de Fairchild, el creador del integrado con el que se armaría la lógica de la computadora, se llegó a un acuerdo para que Texas Instruments y Philco también produjeran estos mismos integrados para no depender de un solo proveedor. En 1963 más de la mitad de todos los circuitos integrados en fabricación se destinaban al armado de los prototipos de esta computadora [1]. El MIT compraba partidas enteras de producción a los fabricantes y si alguno de los componentes no cumplía con los requisitos de funcionamiento, los descartaba y compraba más [2]. Esto dio un gran impulso económico a la industria microelectrónica y ayudó a elevar los estándares de calidad. También contribuyó mucho a reducir los costos de los integrados, que a principios de los ‘60 eran de unos mil dólares por unidad y para 1963, cuando el MIT comenzó a comprar en cantidad, ya se conseguían por sólo 25 dólares.

La computadora de las naves Apolo estaba fabricada con 2800 circuitos integrados [2, 3], todos iguales entre sí y cada uno con un nivel de integración muy bajo: apenas 6 transistores con sus conexiones en cada integrado, lo que da un total de 16800 transistores para toda la computadora. Aun con circuitos integrados tan poco sofisticados la ventaja que se tenía comparado con la utilización de transistores independientes era mucha en ahorro de consumo eléctrico, volumen y conexionado. Pocos años después de iniciado el proyecto del MIT, ya existían más opciones para armar equipos digitales con circuitos integrados. Una muy usada antes de la aparición de los microprocesadores era la utilización de circuitos integrados de la familia 7400, introducida en el año 1964. En esta línea de integrados se dispone de gran variedad de funciones distintas y específicas, como por ejemplo Flip-Flops, operaciones lógicas, registros de giro, buffers, etc. Utilizando estos integrados más avanzados, con decenas de transistores en algunos casos, se puede reducir mucho el uso de componentes.

La ventaja de usar todos componentes iguales, como usó el MIT, se encuentra en que la verificación y manejo de cada uno se simplifica y la complejidad sólo está en cómo conectarlos entre sí. También había habido malas experiencias en otros proyectos que utilizaban variedad de circuitos integrados, como el misil Minuteman II que usaba 19 tipos distintos fabricados especialmente por Texas Instruments, pero donde existieron problemas porque la tecnología de circuitos integrados en la época no estaba preparada para fabricar distintas líneas, cada una con la calidad requerida [4].

En un proyecto tan complejo no se puede estar permanentemente absorbiendo las nuevas innovaciones que van apareciendo porque el trabajo se volvería inmanejable de tantos cambios que habría que realizar. Entonces, en 1966 con la primera versión del Block II, se congeló el diseño para poder tener una computadora funcionando y ya probada. De todas formas la microelectrónica seguiría avanzando y para cuando el programa cumpliera su objetivo de explorar la Luna, en 1969, la tecnología usada en la computadora ya había sido ampliamente superada. En 1972, cuando se realizó la última misión del programa (Apolo 17), ya existían microprocesadores y tecnologías mucho más modernas para todos los componentes de la computadora. De hecho, en los últimos años del programa Apolo quedó un solo proveedor de circuitos integrados para esta computadora, Philco, porque a los otros

ya no les convenía seguir fabricando esta línea obsoleta de componentes para un único cliente.

### Lógica NOR

La electrónica de la computadora estaba formada por miles de circuitos integrados todos iguales entre sí: cada uno era una doble compuerta NOR de tres entradas, similar a un 74HC27 de la actualidad, aunque este último incluye tres compuertas en lugar de sólo dos.

Una compuerta NOR de tres entradas lo que hace es una operación de OR lógico con salida negada. Si tenemos en cada una de las tres entradas un valor que es verdadero o falso (según la tensión que tenga) en un OR lógico la salida es siempre verdadera mientras una de las entradas lo sea, así que el único caso en que la señal de salida es falsa se da cuando todas las entradas son falsas. En este caso, como es un OR lógico con salida negada, esta es verdadera sólo cuando las tres entradas son falsas, el resto del tiempo la salida es falsa.

Las compuertas NOR y las NAND (AND lógico con salida negada) tienen una propiedad que es que con ellas se puede implementar cualquier otra compuerta, lo que las convierte en universales ya que usando estas compuertas se pueden crear compuertas NOT, OR, NAND, AND, XOR o Flip-Flops (Figura 1).

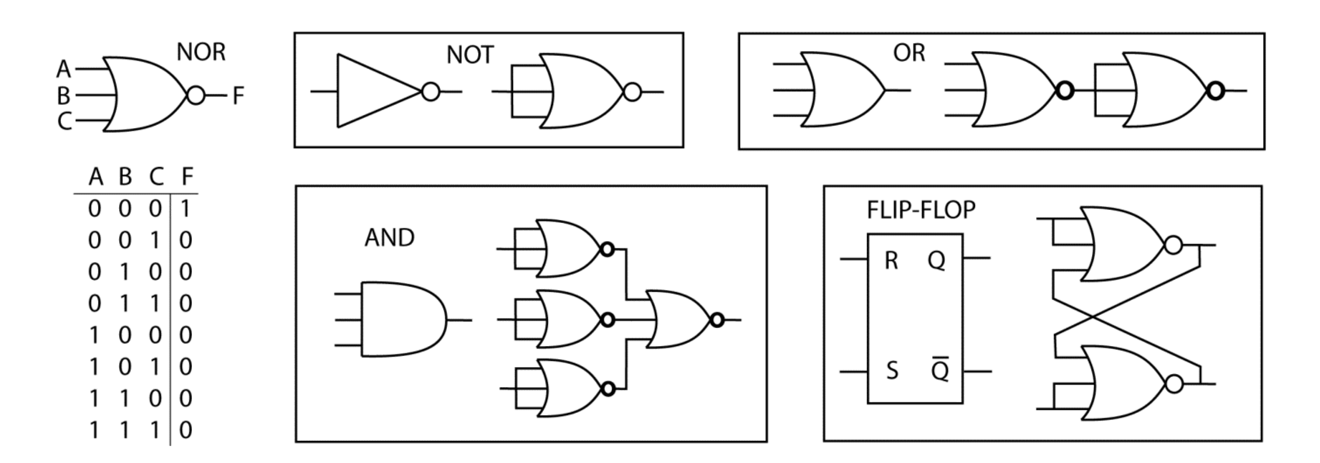


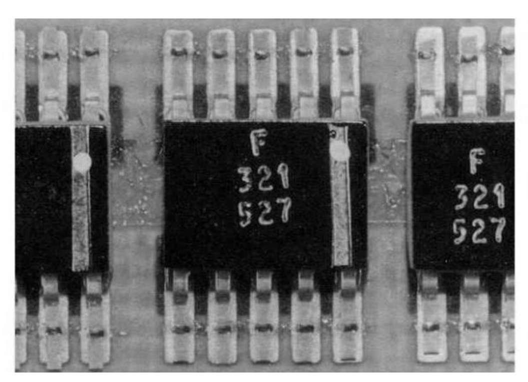

**Figura 1.** Funcionalidad de compuerta NOR y distintas combinaciones para formar otras compuertas.

Esta característica hace que se utilicen en los diseños electrónicos cuando es necesario tener varios tipos de compuertas pero se quiere usar la menor cantidad posible de integrados. Utilizando un triple NOR se puede tener en un solo integrado un negador y un Flip-Flop, si en cambio se utilizara un integrado para el negador y otro para el Flip-Flop se desperdiciaría mucho espacio en la placa.

En el diseño de la computadora se saca provecho a esta propiedad y toda la lógica se implementa con combinaciones de compuertas NOR, así si se necesita un AND, un NOT, o cualquier otro tipo de compuerta lo único necesario para formarlos es agrupar el único tipo de integrado que se posee.

## Encapsulados de circuitos integrados

El Block I se armó con compuertas NOR individuales en pequeñas cápsulas de metal con terminales maleables de varios centímetros (encapsulado TO-47), similares a los encapsulados más comunes que todavía hoy son utilizados para los transistores, pero con 6 patas, dos de alimentación, las tres entradas y una salida [5]. Para reducir más el tamaño y la cantidad de conexiones en la versión Block II se utilizó un novedoso encapsulado más pequeño en donde cada circuito incluía dos compuertas NOR y por fuera tenía diez patas, dos de alimentación y dos pares de tres entradas más dos pares de salidas para las dos compuertas internas. El encapsulado tipo flatpack, introducido en 1963 por Fairchild, ya es más parecido a los encapsulados DIP (dual in line) y a los de montaje superficial (tipo SOIC por ejemplo), en donde el integrado es un rectángulo y las patas cortas y rígidas están ubicadas en dos hileras a ambos lados del integrado. Este componente se ubica sobre un circuito impreso de forma superficial, sin insertar las patas dentro de un agujero a diferencia de la tecnología through-hole (Figura 2).



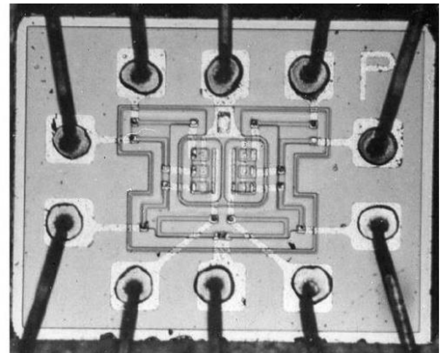



**Figura 2.** A la izquierda se muestra un encapsulado similar al TO-47 usado por los circuitos integrados del modelo Block I. En el centro se ve un integrado de la computadora Block II con encapsulado Flatpack soldado a la placa lógica. A la derecha se puede observar el interior de un integrado Flatpack del Block II con dos compuertas NOR.

El encapsulado de un circuito electrónico es, tal vez, algo secundario comparado con la tecnología existente dentro del circuito en sí, pero la confiabilidad de todo el componente depende en gran medida del encapsulado: por ejemplo la disipación de calor, la resistencia mecánica, la practicidad para soldar en una placa y muchos aspectos relacionados con la interacción con el mundo exterior. La tecnología para crear mejores encapsulados va avanzando junto con la de la microelectrónica, pero a principios de los ‘60 todavía estaba en etapa muy experimental.

## Circuito integrado de amplificación de señales

Los miles de circuitos NOR que formaban la lógica de la computadora no eran los únicos circuitos integrados que había en ella. Para poder leer el contenida de las memorias,

tanto RAM como ROM, hacía falta amplificar las débiles corrientes que se generaban por inducción eléctrica en sus líneas de lectura. Esta amplificación requería de un circuito con varios transistores por señal, y, como se leían en paralelo varios bits (16 de la memoria RAM y 16 de la ROM), hacían falta varios de estos circuitos para amplificar todas están señales a la vez. Además, era importante que esta amplificación se produjera sin que entrara ruido en la señal para que la computadora pudiera leer las memorias correctamente. Para cumplir estos requisitos, y en un espacio limitado, el MIT mandó a Sperry Rand Corporation a diseñar un circuito integrado que realizara esta función. Este integrado era más complejo que las compuertas lógicas al no ser digital sino analógico y era muy importante tener bajo control todas las propiedades de los transistores dentro del integrado para que amplificara la señal de forma correcta, por eso era muy complejo para el estado de la tecnología de la época y terminó siendo el primer circuito integrado amplificador [5].

## ARQUITECTURA DE LA COMPUTADORA

Esta computadora es anterior a la adopción del estándar de representación de las cantidades con 8 bits (un byte) y sus múltiplos (16, 32, 64, etc.) que se fue dando con el tiempo, y en su diseño se optó por usar palabras de 15 bits. Eran comunes en la época computadoras de 36 bits (IBM 7090, PDP-10), 25 bits (UNIVAC III) o 12 bits (PDP-8) entre otras arquitecturas. Este tamaño de palabra se utiliza tanto para datos como para instrucciones.

Las memorias de esta computadora trabajan en realidad con palabras de 16 bits porque cada palabra incluye un bit extra de redundancia. Este bit es redundante porque es dependiente del valor de los otros 15 bits: indica si hay una cantidad par o impar de bits en 1 en los 15 bits de datos y sirve para detectar rápidamente la existencia de un problema de hardware si no coincide con el valor esperado. Con este bit de paridad se puede detectar un problema con una probabilidad de un 50 % en una palabra, pero cuando hay más palabras con errores la probabilidad de detectarlos crece. Este tipo sistema de detección de errores por bit de paridad se solía utilizar incluso en las de PC de escritorio en los ‘80 aunque con el tiempo se fue dejando de lado en la mayoría de los equipos, aunque es útil en un sistema de control en tiempo real poder detener la ejecución de la computadora cuando hay errores importantes, porque de seguir funcionando puede realizar acciones que ponen en riesgo bienes materiales o hasta la vida de personas.

### Elección de palabras de 15 bits

La elección de un tamaño de palabra de 15 bits se hizo analizando las necesidades del programa de navegación y las propias de la computadora. El tamaño de bits de la palabra es el usado para las operaciones aritméticas, como unidad mínima de almacenamiento en memoria (RAM o ROM) y como tamaño de cada instrucción. Mientras más bits pueda manejar a la vez es mejor, tanto para la navegación como para el desarrollo de software,

pero aumenta la complejidad del hardware lo cual se manifiesta en mayor cantidad de circuitos electrónicos, tamaño, peso y consumo eléctrico.

Un tamaño grande de cada palabra, como por ejemplo 32 bits, permite valores de mucha precisión e instrucciones que pueden incluir mucha información, por ejemplo direcciones de memoria grandes que pueden acceder a cualquier lugar del sistema. La contra, además de circuitos más grandes y buses muy complejos con muchas conexiones, es que, con una unidad mínima de memoria tan grande se desperdicia mucho espacio de almacenamiento. El programa en ROM sería muy grande y la mayoría de bits no se utilizarían, porque por lo general no se tiene que trabajar todo el tiempo con cantidades grandes y gran parte de las operaciones son con valores chicos, entonces de los 32 bits en la mayoría de los casos sólo se usarían los bits más bajos y el resto sería casi siempre 0.

En el otro extremo, utilizar palabras más pequeñas, por ejemplo 8 bits, no es práctico en una arquitectura simple como esta porque, como cada instrucción y cada dato tendrían muy poca información, debería realizar cada operación en muchos más pasos, y como se tiene una velocidad de ejecución muy baja y un acceso a memoria lento, la operación de la computadora se vería muy limitada.

Con una palabra de 15 bits se está en un punto intermedio. En cada ciclo la máquina puede procesar números más o menos grandes, aunque para cantidades mayores puede recurrir a usar dos palabras concatenadas (valores de 30 bits). Cada instrucción tiene bastante información, aunque no la suficiente para apuntar a cualquier parte de la memoria pero esto se soluciona utilizando mecanismos externos para poder controlar por partes toda la memoria (ver en *Mapeo de memoria*). El hardware se simplifica bastante a costa de complicar el software que, para poder ver toda la memoria y trabajar con cantidades grandes, debe realizar más pasos.

**Tipos de representación numérica: complemento a uno y a dos**

La computadora trabaja con números de 15 bits, donde se pueden representar valores desde 0 a 32767 si no se tiene en cuenta el signo, pero con signo los valores van desde -16383 a 16383. Para comprender la codificación de números en esta computadora primero hay que repasar algunas de las posibles formas en que se puede realizar. Las computadoras digitales trabajan con valores almacenados en bits, y representar números en estos bits no es algo particularmente complicado: simplemente si se toma a cada bit como un dígito que en vez de ir de 0 a 9, como en el sistema decimal de uso común, lo hace utilizando el sistema binario con valores de 0 o 1 y el resto de las operaciones siguen siendo similares. La situación cambia cuando se tienen en cuenta las cantidades negativas, porque en este caso, aparte de los dígitos que forman al número, hace falta un símbolo extra que represente el signo de esta cantidad.

La forma más directa para indicar si el valor es positivo o negativo es usar el primer bit para indicar el signo (0 positivo y 1 negativo) y los otros para representar el valor absoluto del número. Por ejemplo, usando valores de 4 bits para simplificar, $0101_2$

representa el número 5 y $1101_2$ representa al –5. Utilizando esta representación de las cantidades las operaciones de multiplicación y división no se ven muy afectadas: simplemente usan los bits con el valor absoluto y recién al final se aplica el signo. Por ejemplo para hacer $1010_2$ (-2) multiplicado por $0011_2$ (3) primero se hace $010_2$ (2) x $011_2$ (3) lo que da $110_2$ (6). Luego de hacer la multiplicación se aplica la regla que dice que si los signos son iguales el resultado es positivo (bit de signo en 0) y si son distintos el signo final es negativo (bit de signo en 1), y en este ejemplo el signo será negativo, lo que da el valor $1110_2$ (-6).

Este modo de representar valores ya no resulta tan práctico cuando se realizan las operaciones de suma y resta. Si los dos números a sumar tienen el mismo signo, la suma es directa y no representa ningún inconveniente, pero cuando los signos son distintos ya no se puede realizar una suma, sino que hay que restar un valor sobre el otro, y cuando se resta un valor mayor de uno menor (lo que produciría un cambio de signo) hay que hacer varios ajustes muy complejos en la electrónica, como cambiar el orden de la operación. Por ejemplo $0011_2$ (3) + $1101_2$ (-5) se debería hacer al revés (5-3) y luego cambiar el signo del resultado (de 2 a –2).

Para simplificar esto se utiliza la numeración conocida como complemento a uno, que es similar al sistema de signo y valor absoluto, con la diferencia de que el valor absoluto se representa distinto cuando es negativo: con todos los bits negados. Por ejemplo $0101_2$ representa al 5 y $1010_2$ representa al –5. La ventaja es que el valor absoluto se aumenta siempre hacia arriba cuando se suma, sea el número positivo o negativo. Por ejemplo, de $0101_2$ (5) pasamos a $0110_2$ (6) cuando se le suma 1 y tomando los valores absolutos es $101_2$ (5) y $110_2$ (6). En el caso negativo $1010_2$ (-5) al sumársele 1 se convierte en $1011_2$ (-4) y las secuencias de bits que representan los valores absolutos serían $010_2$ (2) y $011_2$ (3). Como los valores siempre se mueven en el mismo sentido, el circuito es simple y hace sumas y restas sin consideraciones especiales por el signo. Las multiplicaciones y divisiones se realizan igual que en la representación de signo y valor absoluto, salvo que cuando uno de los dos valores es negativo se invierten los bits del valor absoluto.

Un inconveniente que tienen estas dos formas de representar números es que existen dos posibles ceros: un cero positivo ($0000_2$) y un cero negativo ($1000_2$ o $1111_2$ para complemento a uno). Cuando el programa quiere saber si un valor es 0 debe comparar con los dos posibles ceros, y, además, en la lógica electrónica de suma y resta se debe tener en cuenta cada vez que se cambia de signo realizar un paso extra para saltearse uno de los dos ceros.

Esto se soluciona utilizando una tercera forma de representar los números que se llama complemento a dos. Es similar al complemento a uno, con la diferencia de que para hacer negativo un número no se invierten simplemente los bits, sino que se invierten los bits y se suma 1. Por ejemplo –1 se representa como $1111_2$ y –5 como $1011_2$. Con complemento a dos la electrónica es lo más simple posible: ya no existe el cero repetido ni hay que tener consideraciones especiales en las operaciones de suma y resta. La multiplicación y división son ligeramente más complejas en el paso de obtener el valor absoluto de un número

negativo porque tiene que incrementar 1 además de negar los bits. Una inconsistencia que tiene esta forma de representar valores es que, al no haber cero negativo, existe un valor negativo extra que no se puede representar en positivo (por ejemplo en 8 bits se va de -128 a 127 y no se puede representar el 128 positivo), pero es un inconveniente muy menor que no suele importar en el uso normal.

Complemento a dos es la técnica utilizada por todos los microprocesadores de la actualidad, salvo en las unidades de punto flotante que se utiliza una representación con signo y valor absoluto. Esta forma es más compleja porque incorpora también un exponente y además, el circuito está preparado para realizar operaciones mucho más avanzadas que las que hace un procesador de uso general.

**Representación numérica en la computadora Apolo: complemento a uno**

En la computadora de las naves Apolo se utiliza complemento a uno. En la época era un asunto de debate cuál técnica de representación era la más adecuada y todavía no se había adoptado de forma generalizada el complemento a dos como la solución más práctica, y se tenían en cuenta otros detalles que hacían parecer más elegante al complemento a uno: la practicidad en caso de multiplicación y división y la simetría de los valores positivos y negativos. Hay que tener en cuenta que en un inicio la computadora que iría en la sonda a Marte y que fue el comienzo de este diseño, para ahorrar componentes no tendría un circuito sumador completo: las sumas se realizarían en varios pasos independientes en lugar de en un solo ciclo [6] y con un diseño así es poco práctico tener que sumar 1 cada vez que se le quiere cambiar de signo a un número porque se necesitarían varios ciclos de ejecución sólo para realizar este paso tan elemental. Por eso el complemento a dos no resultaba práctico con esa arquitectura y usar complemento a uno parecía algo más natural, aunque los ciclos de suma que se ahorraban al cambiar el signo a un número se perdían al sumar valores de distinto signo. El programa sufriría esta elección por el inconveniente de tener el cero positivo o el cero negativo.

La mayoría de las instrucciones del procesador interpretan todos los números como complemento a uno, no hay como en otros procesadores una versión de las instrucciones para tomar las cantidades con signo o sin signo (por ejemplo en el 8086 existe IMUL y MUL, uno toma dos valores con signo y la otra dos sin signo, siempre positivos). La excepción está en algunas pocas instrucciones incluidas específicamente para trabajar con valores siempre positivos, necesarios para poder representar los ángulos del sistema inercial (ver unidad inercial). Como estos valores se representan como valores enteros, con 32768 valores distintos en lugar de 32767 valores que existen cuando hay dos formas de representar el 0, se necesitan instrucciones especiales para realizar comparaciones entre dos ángulos que utilizan esta representación numérica.

**Números fraccionables**

Esta computadora no cuenta con unidades de manejo de números con punto flotante y las únicas operaciones matemáticas disponibles son las aritméticas básicas que incluye el procesador y siempre trabajan sobre números enteros. Como no hay una forma de manejar números con coma, es el programador el que debe adaptar el manejo de enteros para representar cantidades fraccionadas, analizando cada caso por separado. Por ejemplo, si se quiere representar 1,5 se lo puede hacer guardando el valor multiplicado por un factor de 100, entonces se trabaja con el valor 150 representando así la cantidad fraccionada con una precisión de 2 decimales.

Usando esta técnica se puede sumar y restar fácilmente: por ejemplo 1,5 + 3,2 se realiza sumando 150 y 320 y el resultado es 470 que sacando el factor de 100 es equivalente a 4,7. La complicación viene con la multiplicación y la división: si por ejemplo se quiere multiplicar 1,5 por 2,1, representados en cada caso como 150 y 210, el resultado de multiplicar estos números es 31500 que dividiendo por 100 da 315, pero en realidad el resultado debería ser 3,15 que es 100 veces menor. Esto ocurre porque como los dos valores están multiplicados por 100, al resultado hay que dividirlo por 100 para obtener el resultado correcto. Ocurre, además, que no siempre se utiliza el mismo factor en todas las cantidades, se puede dar el caso de que la velocidad se la represente multiplicada por 1000 pero el tiempo multiplicado por 100, y la distancia por 1000, entonces hay que analizar en cada caso por cual factor dividir el resultado para obtener un resultado usando la unidad correcta. Todo esto es responsabilidad del programador y hace al programa más complejo y lento para ejecutar. Si existiera una unidad de manejo de números con punto flotante esta complejidad pasaría al hardware que maneja los valores fraccionables de forma automática.

**Números de doble precisión**

Utilizando cantidades de 15 bits se tienen 32768 valores distintos (32767 en complemento a 1) y si quisiéramos usar este rango para representar una posición en el espacio entre la Tierra y la Luna, cuya separación es de 380.000 km, habría que dividir esa distancia en unidades del orden de los 10 km. Una unidad mínima así no es nada práctica para establecer donde está la nave en cada momento: esta precisión es mucho menor que la que pueden alcanzar los instrumentos de navegación. Utilizar una palabra de 15 bits no serviría para representar esta cantidad ni muchas otras utilizadas en la navegación.

En el MIT decidieron que el tamaño de los valores con el que convenía trabaja cuando se necesitaban valores de mayor precisión sería de 28 bits [7], lo que permite valores de 0 hasta aproximadamente 200 millones. Utilizando dos palabras de 15 bits, cada una con una cantidad con signo representada como complemento a uno, como los dos números tienen signo se usan 28 bits para representar la magnitud y 2 bits para el signo, lo que es bastante extraño porque lo común es tomar como bit de signo el más alto del conjunto y todo el resto como magnitud, pero para simplificar en el uso de las pocas instrucciones de

manipulación de números se terminó adoptando este formato que desperdicia 1 bit. Normalmente sólo debería importar el signo de una de las palabras y el otro se ignoraría, lo que daría un valor de 29 bits con 1 de signo y 28 de magnitud, pero según la forma en que se hagan las cuentas pueden divergir los dos signos y una de las palabras ser positiva y la otra negativa. Trabajando con valores de esta cantidad de bits la distancia de 380.000 km se podría representar con una precisión de 1 metro, por lo cual ya tiene una mayor utilidad.

Cuando se utilizan cantidades representadas por más de una palabra se debe elegir que parte del número le corresponde a cada palabra. En la numeración decimal se escriben las cifras de mayor valor a la izquierda y las de menor a la izquierda, en orden decreciente de lectura. En computación se puede guardar en la primera dirección la palabra de mayor valor seguida de la de menor (llamado big-endian) o al revés, comenzando con la palabra de menor valor (little-endian). En esta computadora se optó por usar big-endian pero en muchas otras arquitecturas, por ejemplo el 8086 de Intel, se eligió la forma opuesta.

**Tecnología de las memorias**

La computadora cuenta con dos memorias: la RAM de 2048 palabras de 16 bits (15 de datos y 1 de paridad), lo que equivale a 4 kB, y una memoria ROM de solo lectura con 36864 palabras o 72 kB. Ambas memorias se conectan al mismo bus de direcciones y se puede acceder indistintamente a una o la otra. La característica principal de ambas memorias es que se puede acceder a cualquiera de sus palabras en cualquier momento sin seguir un orden específico, pero conectar cada uno de los miles de bits a la computadora utilizando miles de cables y conexiones sería una estructura inmanejable. Una forma de agrupar los bits es por medio de grillas, donde seleccionando una fila y una columna se accede a un bit en particular: así con 32 señales de columnas y otras 32 de filas se pueden direccionar 1024 bits (o sea 32 * 32), y los distintos bits que forman la palabra a acceder (en este caso 16) se obtienen utilizando 16 grillas en paralelo, todas usando la misma dirección (fila y columna) pero trabajando sobre distintas unidades de almacenamiento, cada una conectada a una de las 16 señales del bus de datos (Figura 3).

Para almacenar la información dentro de cada bit se pueden utilizar distintas tecnologías, pero la más obvia es el flip-flop, que es el tipo de almacenamiento que, por ejemplo, se utiliza en todos los circuitos lógicos que necesitan guardar información. En el caso del procesador y circuitos adyacentes hay una gran cantidad de flip-flops, cada uno con una función específica, por ejemplo para almacenar el resultado de una cuenta o para guardar la dirección de ejecución del programa. El flip-flop se conecta directamente a otras compuertas lógicas y no necesita ningún circuito que lo administre. Una memoria RAM constituida por flip-flops es llamada memoria estática y, a pesar de ser fácil de usar y rápida, su principal contra es que cada bit está formado por varias compuertas y por eso no es la forma más práctica de almacenar una gran cantidad de datos.

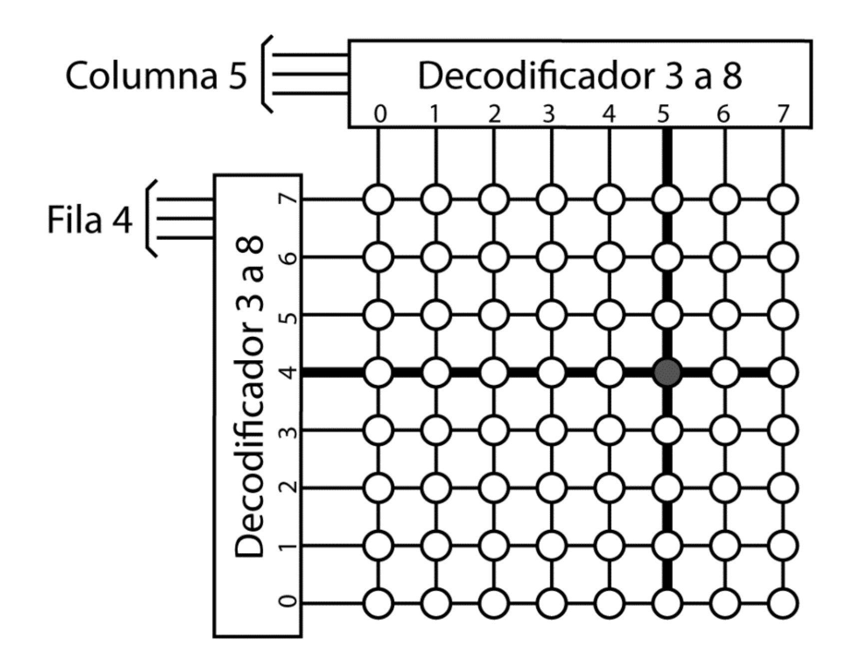



**Figura 3.** Una memoria muy básica con 64 bits en una grilla de 8 * 8. Con 6 bits, 3 para seleccionar la columna y otros 3 para la fila se selecciona un bit en particular.

Desde la década del 1970 la forma de almacenar la mayor parte de la RAM es con memoria dinámica, donde cada bit se almacena en un pequeño capacitor y uno o dos transistores conectan cada bit a la grilla. Este tipo de memoria tiene la ventaja de ser más simple de implementar y de que entran muchos más bits en un mismo integrado que usando una memoria estática, pero son más lentas y necesitan circuitos dedicados para refrescar la información de cada capacitor porque pierde su carga en un breve lapso de tiempo si no se lo actualiza. En el caso de la memoria de solo lectura, una ROM común en un circuito integrado se implementa como una grilla de fusibles que representan el valor de cada bit según si conduce corriente o fue quemado y se accede a cada uno por medio de uno o dos transistores.

Si tenemos en cuenta que entre ambas memorias existían más de medio millón de bits, aunque pudieran utilizar un solo transistor por bit igual existiría una cantidad de semiconductores para las memorias mucho mayor a los usados en el resto de la computadora. Con la tecnología de semiconductores de la época, estas memorias serían muy grandes y consumirían mucha energía al tener que alimentar a cada bit. Ninguna computadora de la época utilizaba memoria con semiconductores y no sería una ventaja hacerlo en este equipo, pero existían alternativas, algunas ya ampliamente probadas en la industria para cumplir esta función.

### Memoria RAM

La tecnología de RAM utilizada en la computadora de las naves Apolo era la de núcleos magnéticos [8], que era el tipo de memoria más común en los '60. En esta memoria cada bit se representa con un muy pequeño anillo (núcleo) de ferrite, un material que es

fácil de imantar y que luego mantiene este estado (Figura 4). Cuando este núcleo está imantado en una dirección representa un 1, y en la dirección opuesta un 0, de la misma forma en que se lo hace en un disco magnético, con la diferencia de que aquí cada bit es un elemento distinguible a simple vista. Otra diferencia con un disco es que no se utilizan medios mecánicos para acceder a cada bit, como el cabezal que viaja físicamente hacía donde está el bit, sino que se accede a cada núcleo por medio de un cableado en grilla, aprovechando las propiedades del material magnético. La alineación magnética se puede cambiar aplicando un campo de una intensidad que supere un umbral mínimo y para esto lo que se hace es disponer los núcleos en una grilla donde a cada uno se le cruzan dos cables, uno correspondiente a la columna y otro a la fila. Cuando se quiere escribir un valor en un bit en particular, se pasa la mitad de la corriente necesaria para cambiar la polaridad del ferrite por el cable de la columna correspondiente y otra mitad por el cable de la fila donde está el bit que deseamos cambiar. En cada cable por el que circula una corriente se genera un campo magnético a su alrededor pero en los ferrites por donde circula corriente en solamente uno de los dos cables (de columna o fila) el campo magnético generado no tiene la intensidad para cambiar la polaridad del material. Únicamente un núcleo será alterado: el que reciba el campo magnético completo, que es el que está en la intersección de los dos cables activos.

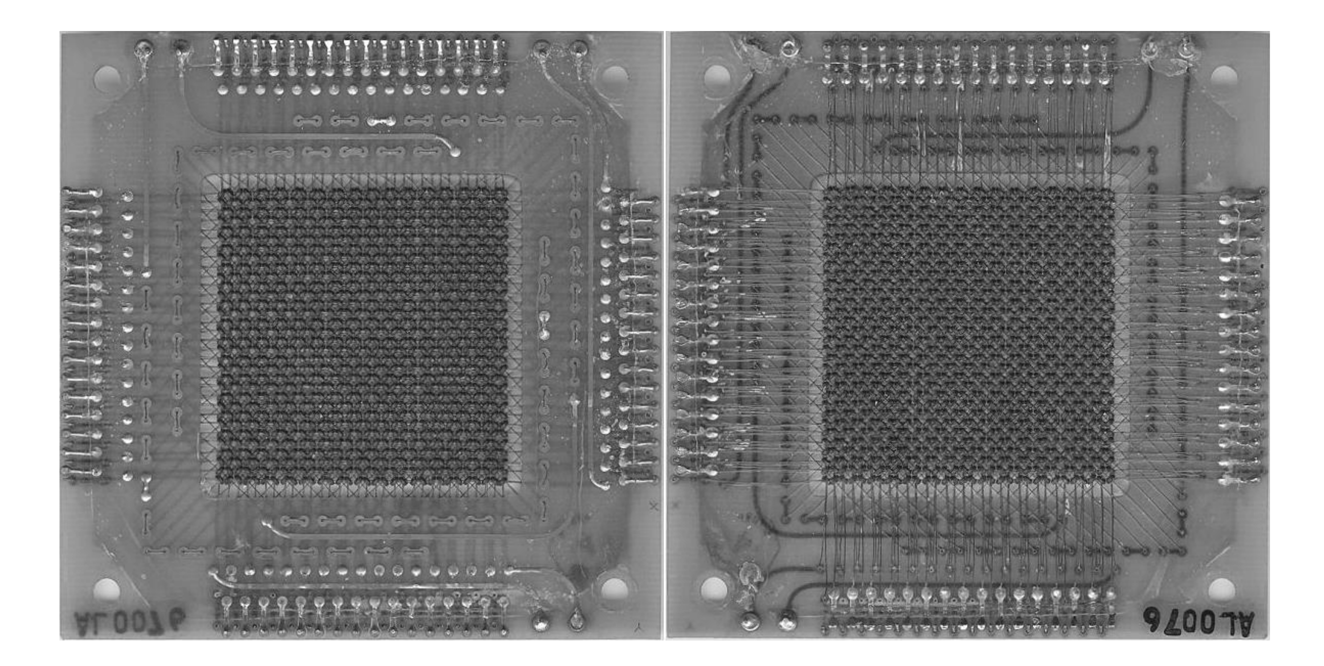



**Figura 4.** Módulo de memoria de núcleos magnéticos de la computadora las naves Apolo. Fotografiada de ambos lados contiene una grilla de 32 * 32 con 1024 bits (foto de Jonathan Ward)

La lectura de un bit es un proceso destructivo, porque lo que se hace es escribir un valor (0 o 1) en el bit a leer. Si el valor anterior del ferrite era opuesto al nuevo, entonces se cambia la polaridad en este instante y este cambio a su vez induce eléctricamente un pulso sobre una línea que pasa por los núcleos y al leer o no un pulso sobre esta línea se deduce el valor que había almacenado en ese núcleo. El siguiente paso en la lectura es tomar ese valor obtenido y reescribirlo en el núcleo para no perder la información en próximas lecturas.

Para administrar la grilla y activar las columnas y filas hacen falta circuitos dedicados, y para manejar la secuencia de lectura destructiva hace falta ocupar varios ciclos a cada lectura y guardar el resultado en buffers intermedios. Todo este proceso para leer o modificar una palabra le lleva a esta computadora 12 pasos distintos que realiza cada uno en 1 µs (Figura 5).

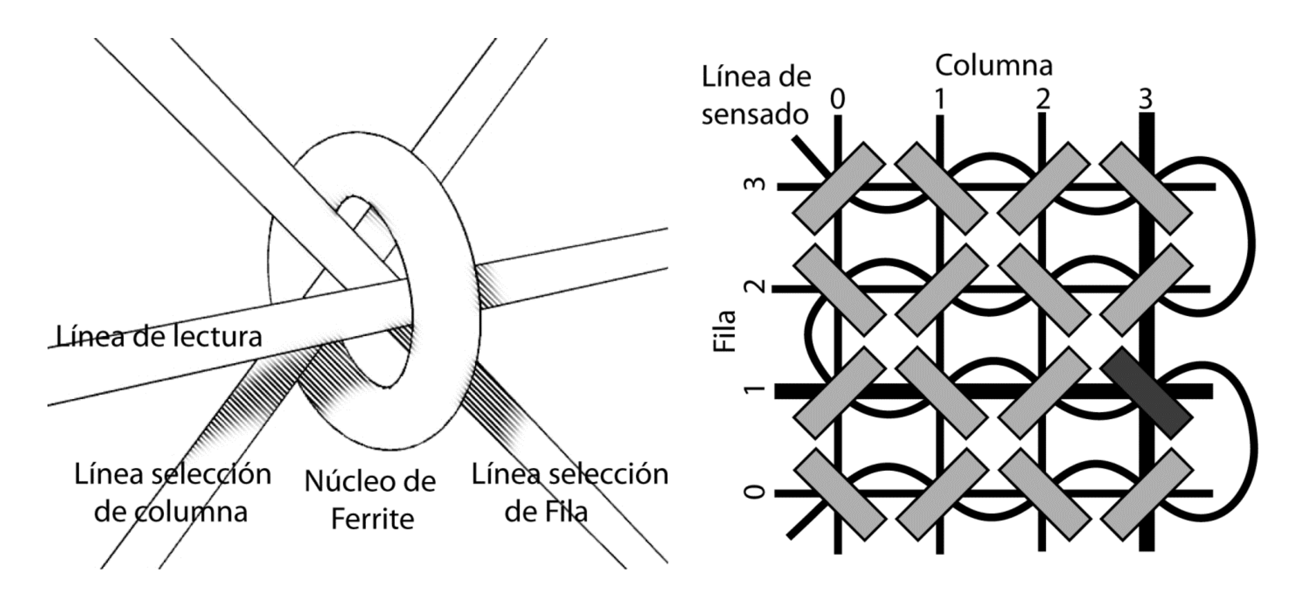



**Figura 5.** A la izquierda el detalle de un núcleo con las dos líneas de selección y escritura y la de lectura. A la izquierda una grilla de 16 bits (4 * 4) donde se escribe un valor en el bit de la columna 3 fila 1 enviando corriente a las líneas correspondientes. La línea de sensado pasa por todos los núcleos.

El uso de estas memorias tiene varias ventajas con respecto a las RAM estáticas o dinámicas: el principal es que no necesitan una alimentación eléctrica constante porque, al igual que un disco, almacena la información magnéticamente y esto ayuda a reducir el consumo de energía de la computadora y hace al sistema más robusto, menos sensible a ruidos en la alimentación que en una RAM dinámica o estática pueden corromper la información. Otra ventaja para la utilización en el espacio exterior es su completa resistencia a la radiación: los circuitos integrados son susceptibles de sufrir alteraciones al ser irradiados por partículas cargadas eléctricamente y otros efectos de largo plazo de la radiación, pero, al menos en la cercanía de la Tierra, no hay en el espacio un campo magnético tan intenso como para alterar el contenido de una de estas memorias.

## Memoria ROM

La memoria ROM de esta computadora es mucho más grande que la RAM (18 veces mayor) y por eso necesita almacenar la información de forma mucho más compacta para no ocupar un espacio excesivo. Los núcleos magnéticos permiten modificar la información y en esta memoria no es necesaria esa funcionalidad y, de hecho, es indeseable. Lo que se utilizó fue una memoria de soga (Core Rope Memory) [3], donde se usan también pequeños anillos

o núcleos de metal, pero de forma distinta a la memoria de núcleos magnéticos [9, 10]. Aquí no se guarda información de forma magnética y tampoco por cada núcleo se almacena solo un bit sino 192. Una soga formada por 210 pequeños cables aislados entre sí pasa por 512 anillos de metal y, en cada núcleo, algunos cables van por afuera y otros por adentro del núcleo. 192 de esos cables son los que guardan la información, los bits, en cada intersección de cada cable con un núcleo: si un cable pasa por afuera el valor del bit en esa intersección es 0, en cambio si pasa por adentro representa un 1. Para leer el contenido de una dirección de memoria se envía un pulso a un núcleo en particular y este pulso induce una tensión en las líneas que pasan dentro de él, entonces de los 192 cablecitos los que pasan dentro del anillo producen un pulso, similar a como ocurre en un transformador en el bobinado secundario, y los cables que pasan por afuera no producen nada. Leyendo cuál de las 192 líneas tienen tensión se obtiene si cada línea representa un 1 o un 0 si no produce nada. Como sólo se necesita leer una palabra de 16 bits, un circuito selecciona unicamente 16 entradas de las 192 según la dirección que se está leyendo (Figura 6).

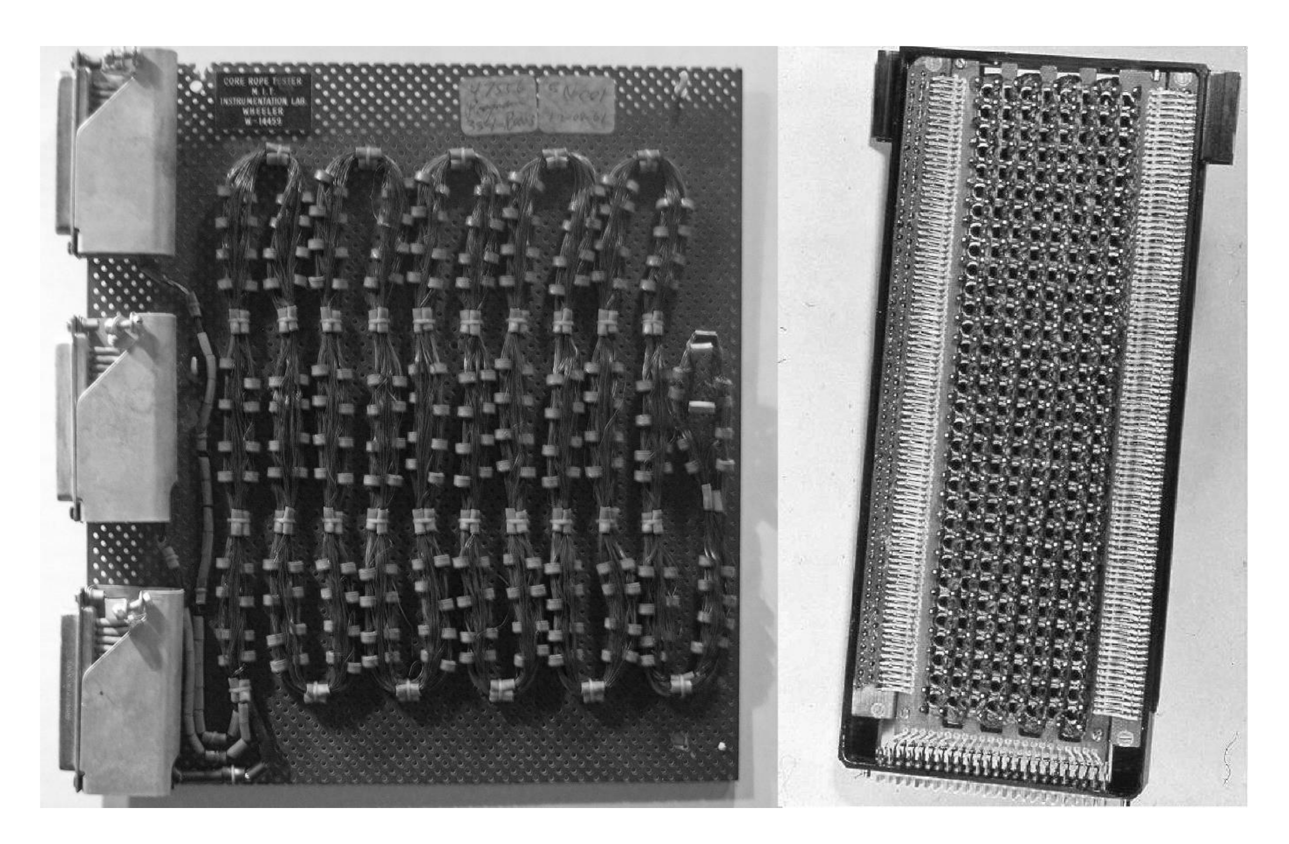



**Figura 6.** A la izquierda un prototipo de memoria de soga y su lado un módulo terminado.

Para generar un pulso solamente en un núcleo y no en los 512, y así sensar el cableado en ese núcleo, se recurre a los otros 18 cables de la soga. De un grupo de 4 cables se envía un pulso a uno sólo que pasa por 128 núcleos y así se limita una cuarta parte de los núcleos de la soga. Para que el pulso induzca una corriente en uno solo de esos 128 núcleos se utilizan los otros 14 cables de selección divididos en 2 grupos de 7. Estos cables son las líneas de inhibición y ahí se carga una dirección binaria de 7 bits (para seleccionar un núcleo entre 128) y en los otros 7 cables se envía la misma dirección pero invertida. En cada nodo

pasan siempre 7 de estos cables, y por cada una de esas señales se toma o el cable con el valor directo o con el valor invertido, realizando las 128 posibles combinaciones en cada nodo. Entonces, al enviar un valor binario de 7 bits por las líneas de inhibición en sólo uno de los 128 nodos habrá una combinación de líneas de inhibición que tienen todos los valores iguales y de misma dirección que la línea de pulso común a todos los nodos. En todos los demás nodos pasa al menos una señal de inhibición y entonces no se induce una corriente en estos núcleos, y de esta forma se garantiza que se lea un sólo nodo de toda la soga. Cada soga guarda 98304 bits (192 * 512) lo que dan 6144 palabras de 16 bits. En total había seis módulos de estas memorias para llegar a las 36864 palabras (Figura 7).

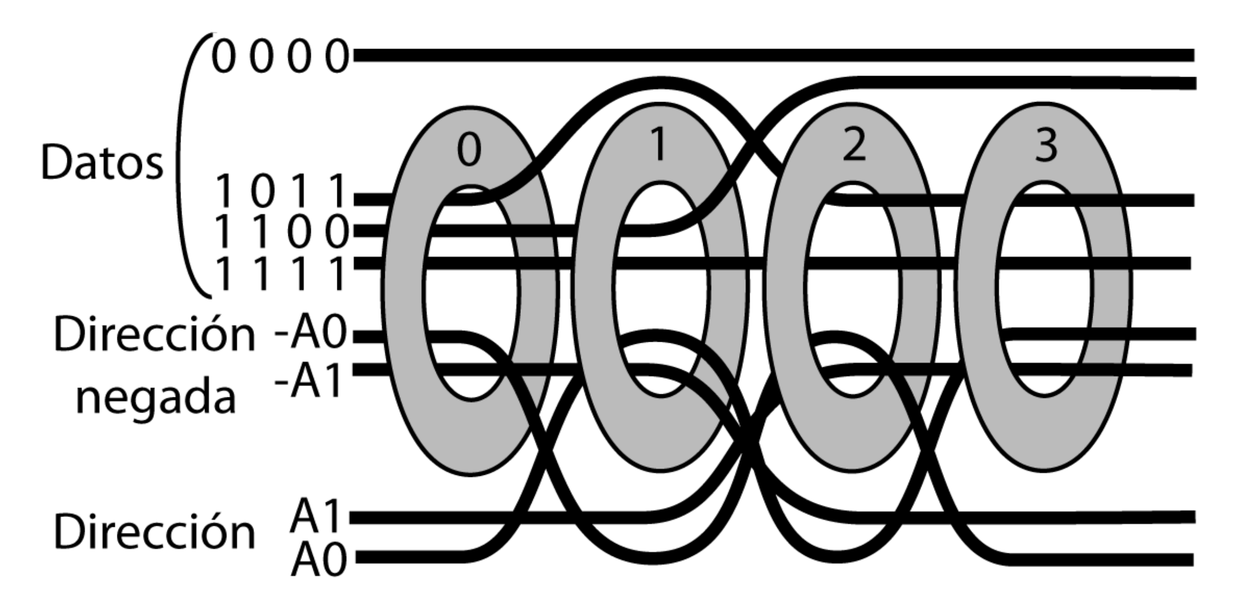



**Figura 7.** Ejemplo simplificado de memoria con 16 bits, 4 direcciones con 4 bits en cada núcleo. Las líneas de datos guardan la información según su posición dentro de cada núcleo. Las 2 líneas de dirección y sus complementos activan un solo núcleo por vez que genera un pulso en las líneas de datos dependiendo de su posición.

Esta forma de almacenar la información para la época era muy eficiente en cuanto al espacio, pero una contra muy importante es el tiempo que llevaba grabar los datos con esta técnica. Una memoria ROM común se fabrica en blanco, con todos los bits en 1 y luego se puede acceder a cualquier bit y pasarlo a 0, y así el proceso completo de grabación dura segundos. En una memoria de soga el valor de cada bit se determina en el momento de su fabricación: no es posible fabricar memorias en blanco y luego grabarles la información. En la fábrica literalmente se tejían estas memorias pasando minuciosamente cada cable por dentro de un anillo o por afuera según correspondiera. Cada cable muy finito debía pasarse por múltiples núcleos y para hacer esto era necesario ser muy cuidadoso para no romper un cable tan delicado. En cada núcleo pasaban siempre 8 cables, los de selección, y luego una cantidad variable de hasta 192 cables que normalmente eran sólo 96 cables por núcleo, que era el promedio de 1 s en las palabras guardadas ahí. Por la organización de la memoria, las 12 palabras que se guardan en cada núcleo no eran consecutivas dentro del espacio de direccionamiento sino que estaban separadas por 256 direcciones, por eso era altamente

improbable que tuvieran valores en común y todas tuvieran los bits en 1. Si esto hubiera ocurrido habría sido un problema importante porque no entraban todos los 192 cables dentro del núcleo y esa información no se podría almacenar. Informalmente, la técnica de creación de estas memorias se la llamaba *Little Old Ladies* (de las *viejitas*) [11] porque era un grupo grande de costureras que, siguiendo instrucciones precisas de una computadora, debían armar estas sogas. Las primeras versiones eran literalmente sogas con los núcleos en el medio (Figura 8). La versión final, que volaría a la Luna, estaba mejor organizada en una grilla pero siguiendo el mismo principio.

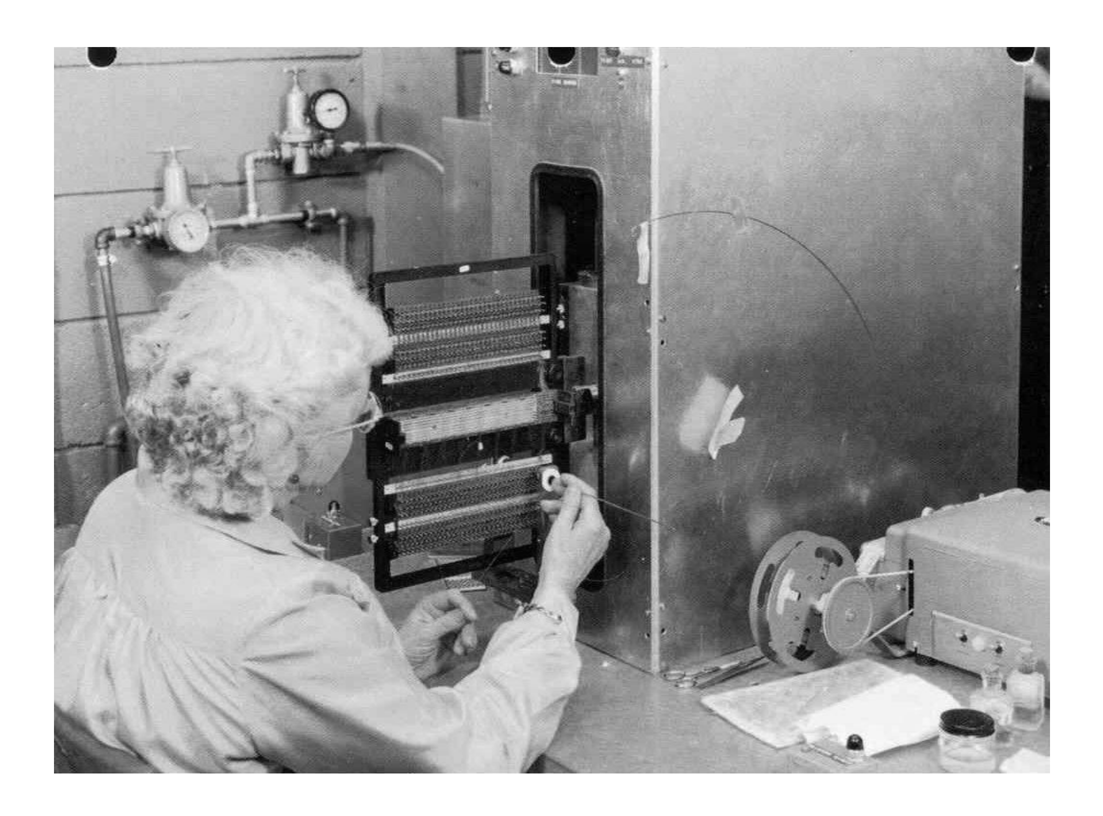
Una enorme limitación que tenía esta fabricación era la anticipación con la que se debía tener terminado y probado el programa que se usaría en la misión antes de cada despegue. Se necesitaban aproximadamente 4 meses para fabricar las memorias de forma confiable, y en ese tiempo no había forma de alterar el programa: los datos que debían grabarse en la memoria tenían que estar definidos al comenzar a fabricar la memoria. Esto contrasta fuertemente con los desarrollos más modernos donde el software se actualiza incluso después de entregado el producto. En misiones espaciales más modernas, como el robot Curiosity en Marte, el software de navegación grabado en una memoria Flash se actualiza incluso durante el viaje.

Que el programa necesitara terminarse con tanto tiempo de anticipación parecía contradecir el hecho de tener un sistema programable porque en la plataforma de despegue los programadores no podrían modificar el programa y le quitaba mucha flexibilidad. Pero, por otro lado, ayudaba a reforzar la disciplina en el desarrollo de software: los programadores estaban forzados a terminar el programa con anticipación y no podían posponer detalles para último momento. Esta computadora no contaba con ningún otro medio de almacenamiento, ni disco ni cintas magnéticas de ningún tipo, y lo que no se incluyó en la ROM meses antes de lanzar la misión sólo se podría actualizar de forma muy limitada en la RAM.

## CARACTERÍSTICAS GENERALES DE LA COMPUTADORA

Dos requerimientos muy importantes de la computadora eran los de peso y de volumen. El espacio dentro de las naves es muy limitado, tanto para los astronautas, que están muy restringidos en sus movimientos en un espacio tan pequeño, como para todos los instrumentos y dispositivos que deben incluir dentro de la nave. Una parte muy importante del diseño de estos vehículos es buscar cómo aprovechar de la forma más eficiente el poco volumen disponible para que entre todo lo que se necesita para poder llevar a cabo una misión a la Luna. Como el diseño de las naves se realizaba en una etapa en que también se estaba trabajando en el diseño de la computadora y no se contaba con especificaciones precisas que indicaran cuál era el volumen este equipo, se debía trabajar con estimaciones. North American, el fabricante del Módulo de Comando, la primera nave que se diseñó, desde el comienzo definió el volumen que podía ocupar la computadora en el panel en 1 pie cúbico (unos 30 mil $cm^3$). El diseño de la computadora tuvo como meta no excederse de ese

volumen y se aplicaron las mismas dimensiones para la computadora del Módulo Lunar que era similar.



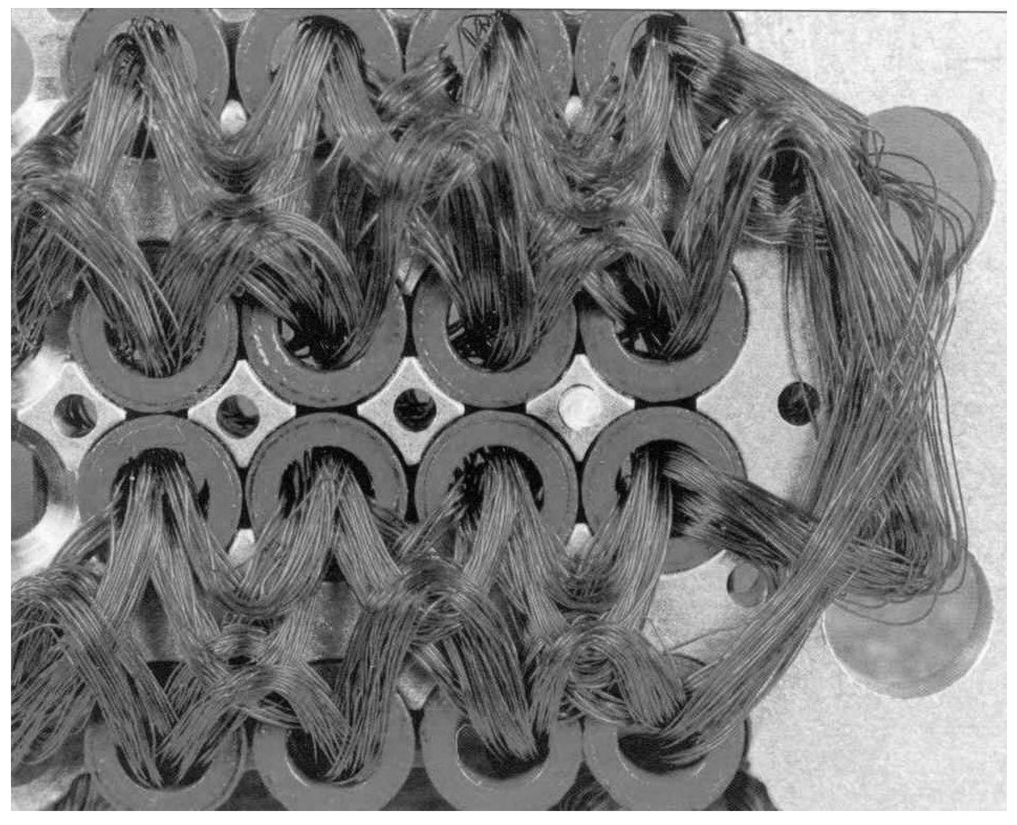


**Figura 8.** Arriba, una operadora tejiendo el programa en la memoria ROM. Abajo, una ampliación de varios núcleos con su cableado.

El peso que tendría la versión final sería de 32 Kilos y las dimensiones del gabinete eran de 61 x 32 x 17 cm [3].

## Velocidad de ejecución

La frecuencia del reloj de referencia es de 2 MHz, y de él se obtienen distintas frecuencias usadas para los distintos relojes que usa la computadora, entre ellos el reloj principal del procesador que es de 1MHz [3] y es en base a este reloj que se decodifican las instrucciones de la computadora, aunque en un ciclo no se ejecuta una instrucción entera sino que se requieren múltiples ciclos para procesar cada una. Lo que determina la velocidad de todo el sistema es la memoria RAM de núcleos magnéticos que requiere 12 pasos para leer una palabra porque primero tiene que escribir un valor fijo para poder leer y luego debe reescribir el valor leído (lectura destructiva). Aunque se lea de la memoria ROM, que no necesita tantos pasos, o de un registro interno que son más veloces o mismo una escritura a la RAM, que requiere menos pasos que una lectura, la duración del ciclo de lectura es lo que limita todo el funcionamiento de la computadora. Este ciclo de 12 µs, llamado ciclo de memoria, es la unidad mínima de ejecución del procesador.

Si todas las instrucciones fueran simples y no trabajaran sobre otras posiciones de memoria, entonces todas se ejecutarían en 12 µs y por cada segundo se podrían ejecutar más de 80.000 instrucciones, pero la mayoría de las instrucciones trabajan sobre otra dirección más de memoria así que corren el doble de lento (24 µs) y algunas dedicadas a trabajar con números grandes acceden a dos direcciones de memoria, lo que da un tiempo de ejecución de 36 µs por cada una. Si una instrucción debe realizar un procesamiento independiente de la memoria, por ejemplo una operación aritmética, el tiempo extra también será un múltiplo del ciclo de memoria de 12 µs.

## Consumo de corriente

El consumo de la computadora era de 55 W [3] y necesitaba para funcionar 4 V en los circuitos lógicos y 14 V en los circuitos de manejo de memorias y, además de eso, se requerían distintas tensiones para los varios periféricos más todo el instrumental eléctrico de la nave. La alimentación eléctrica en las naves Apolo venía desde baterías de óxido de plata y celdas de combustible. La celda de combustible genera energía eléctrica combinando oxígeno e hidrógeno y como resultado, además de energía eléctrica, se obtenía agua a la que se le daba distintos usos en la nave, por ejemplo para beberla [12]. Esta celda de combustible solamente estaba en el Módulo de Servicio y alimentaba al Módulo de Comando y era la principal fuente de energía eléctrica durante el tramo más largo del viaje, porque todo el sistema con los tanques de oxígeno e hidrógeno era mucho más liviano que el equivalente en baterías. En el Módulo Lunar, por simplicidad, sólo se disponía de baterías como fuente de energía, lo mismo en el Módulo de Comando, que una vez que se desprendía del Módulo de Servicio, contaba con baterías para alimentar los sistemas

eléctricos durante la reentrada a la atmósfera. En caso de rotura de la celda de combustible (como ocurrió en la Apolo 13) la energía de las baterías debía usarse con mucho cuidado porque no alcanzaba para mantener encendidos todos los dispositivos durante mucho tiempo.

Siempre que se realiza un trabajo físico se genera, a la vez, calor en mayor o menor medida según la eficiencia con que se trabaje, y el caso de los sistemas eléctricos no es la excepción. En las computadoras y circuitos digitales, por ejemplo, casi la totalidad de la energía consumida termina generando calor porque el trabajo consiste en mover pequeñas cargas dentro de los circuitos, todo el resto de la energía es desperdiciada y, como en la década de 1960 la electrónica era mucho menos eficiente que en la actualidad, el calor generado era mucho mayor. En el espacio es muy complicada la disipación del calor, porque, al no haber una atmósfera, no es posible hacer circular aire y eliminar el calor excesivo al exterior: todo el aire disponible es el poco que hay en la nave, y la disipación por radiación en el vacío es muy limitada. Incluso los satélites y sondas espaciales más modernos deben reducir el consumo eléctrico lo más posible para evitar estos inconvenientes, y, por ejemplo, no usan el microprocesador más rápido que haya disponible, sino que se opta por el más eficiente y se lo usa sólo a la velocidad necesaria.

Las naves Apolo cuentan con un complejo sistema de control de temperatura que, entre otras cosas, dispone de cañerías con agua que se hacen circular alrededor de los sistemas eléctricos. Solamente para casos puntuales existían en las naves dispositivos para generar calor, por ejemplo para manipulación de los gases licuados de los tanques de oxígeno e hidrógeno. El resto del sistema de control de temperatura estaba diseñado para deshacerse del calor excesivo generado por los dispositivos eléctricos. De hecho se contaba con la existencia de este calor residual para mantener la temperatura en valores aptos para los tripulantes. Sin la mayoría de los sistemas eléctricos encendidos se corría el riesgo de que, a medida que se irradiaba lentamente el calor de la nave al vacío, la temperatura adentro se volviera muy baja.

La computadora dispone de un modo de bajo consumo, o standby, que es activado por una secuencia de teclas y lo que hace es cortar la alimentación de la computadora excepto la parte que genera señales de reloj. A su vez, estas señales de reloj generan periódicamente (cada 10 ms) una interrupción en la computadora para que incremente el contador con la hora (ver interrupciones) y cuando ocurre este evento la computadora se vuelve a activar para que pueda mantener el reloj actualizado. Este tipo de operación en modo de bajo consumo es muy común en los microcontoladores modernos y funcionan de forma similar. De todas formas, esto no se llegó a usar, porque inicialmente se pensó en dejar la computadora apagada durante tramos largos del viaje. Sin embargo, luego se dejó de lado y la computadora pasó a estar encendida todo el trayecto para no perder la orientación de la nave y tener que recalibrarla.

## Instrucciones del procesador

La tarea que realiza el procesador de una computadora es ejecutar en orden las instrucciones que hay en la memoria. Existe un registro de la computadora que tiene la dirección en la que se encuentra la próxima instrucción a leer y ese registro aquí se llama Z, aunque en otras arquitecturas por lo general se llama IP (Instruction Pointer) o PC (Program Counter). El procesador lee el contenido de la memoria en la dirección apuntada por Z y decodifica ese valor para obtener la operación que debe realizar y un valor con el cual realizar esa operación: por ejemplo, una operación puede ser sumar (llamada AD en esta arquitectura) que viene acompañada de la dirección de memoria donde hay un valor a sumar. Luego de ejecutar la acción indicada, se incrementa el registro Z y se lee otra vez la memoria pero ahora se saca la siguiente instrucción, y así se van ejecutando todas las instrucciones del programa.

Las instrucciones forman parte de lo que se llama lenguaje de máquina, el lenguaje que entiende una computadora en particular y que a diferencia de los lenguajes de alto nivel como C, Pascal o Basic, está pensado para implementarse de la forma más simple y eficiente en hardware con compuertas lógicas, y no para ser fácilmente entendible por humanos. Una instrucción de esta computadora ocupa una palabra (15 bits) aunque en otros equipos puede tener una cantidad variable de palabras, y así una instrucción que no necesita parámetros (por ejemplo para hacer negativo a un valor) ocupa un solo Byte, en cambio una instrucción que usa un parámetro (por ejemplo sumar un valor de 8 bits a un registro) ocupa dos Bytes. En este caso, como con cada palabra que se lee se tiene una instrucción, se simplifica el ciclo de ejecución de instrucciones al ser igual para todos los casos (Tabla 1).

En los 15 bits de una palabra se codifica en los 3 bits superiores la operación a realizar (por ejemplo, sumar), lo que permite 8 tipos de instrucciones distintas, y los otros 12 bits son una dirección de memoria para usar junto con la operación (en la suma, por ejemplo, apunta a un valor a sumar). Esta dirección es un valor de 0 a 4095, lo que resultaba útil en las primeras versiones de la computadora que contaban con muy poca memoria, pero en el modelo final no permite identificar a una de las 38.000 posibles direcciones. Para superar esta limitación y poder utilizar todas las direcciones se debe recurrir a un circuito externo al procesador, el mapeador de memoria, que expande la capacidad de direccionamiento pero de forma indirecta: el procesador en sí en cada momento sólo puede ver un 10% del total de la memoria pero el mapeador selecciona cuál es ese subgrupo de direcciones.

| Operación | Parámetro |
|---|---|
| 14 13 12 | 11 10 9 8 7 6 5 4 3 2 1 0 |

**Tabla 1.** Formato de una instrucción de lenguaje de máquina

Las 8 posibles instrucciones que permite este diseño también resultaban útiles en las primeras versiones, pero a medida que se le fue agregando funcionalidades al equipo esta cantidad quedó muy limitada, por eso se recurre a distintas técnicas para permitir más operaciones sin salir de este formato. Una forma en que agregan más operaciones es según el área de memoria a la que apunta la instrucción: memoria RAM o ROM. Como estas memorias están en rangos de direcciones bien definidas, se puede, por ejemplo, utilizar un código de operación que cuando trabaja sobre RAM realiza una operación que necesita modificar esa dirección de memoria pero cuando lo hace sobre ROM realiza una operación distinta pero de lectura. También se separan combinaciones especiales de una operación y una dirección en que esa operación que no tienen ninguna utilidad (por ejemplo saltar a ejecutar en una dirección que no tiene código ejecutable) y se arman instrucciones especiales que no necesitan parámetros, por ejemplo para habilitar o deshabilitar las interrupciones.

Una instrucción sin parámetros es EXTEND, que lo que hace es dejar marcado un indicador para que la próxima instrucción se interprete como una operación extendida, lo que significa que el código de 3 bits de la siguiente instrucción se ejecuta, no como una de las 8 posibles acciones de las instrucciones normales, sino como una de las 8 acciones extendidas. De esta forma se puede ampliar aún más la cantidad de instrucciones aunque las que necesitan de EXTEND terminan ocupando más de una palabra en memoria y tardan un ciclo más en ejecutarse, pero para el procesador cada palabra sigue siendo una instrucción individual. En total, con todas las extensiones que dispone, el procesador permite 34 instrucciones distintas, entre las 8 básicas, las 8 extendidas con EXTEND, 7 instrucciones de entradas salidas (ver más adelante) y otras 11 combinaciones especiales.

**Instrucciones aritmético – lógicas**

Las instrucciones que realizan operaciones aritméticas (sumas, restas multiplicación o división) o lógicas (operaciones binarias como el AND) lo hacen sobre un registro especial llamado el acumulador o A, que en la mayoría de los procesadores existe con ese mismo nombre. Se cuenta con una instrucción (CA) que carga en A un valor que saca de una dirección de memoria y también existe la operación inversa (TS), que guarda el valor de A en una dirección de memoria. Una vez que se tiene un valor en A se pueden realizar operaciones matemáticas sobre él, por ejemplo se le puede sumar un valor con la instrucción de suma (AD) que apunta a una dirección de memoria donde hay un valor para sumarle. De forma similar se le puede restar un valor a A con la instrucción SU, o realizarle un Y lógico bit por bit con otro valor de 15 bits usando la instrucción MASK. Existen otras operaciones para realizar sobre A, como cargar un valor desde memoria pero con el signo cambiado (CS), intercambiar el valor en A con un valor en memoria (XCH) o realizar una suma pero agregándole el valor de A directamente a una dirección de memoria (ADS).

Además de las instrucciones que realizan operaciones con un valor de 15 bits en A, existen otras para trabajar sobre valores de doble precisión (30 bits en dos palabras). En este

caso, además del registro A se utiliza otro registro llamado L en donde están los otros 15 bits del valor de 30 bits: en A están los 15 bits superiores y en L los más bajos. Para cargar un valor de 30 bits en A y L se usa la instrucción DCA que saca el valor de dos palabras consecutivas en memoria y para sumar A y L a la memoria (dos palabras) se utiliza DAS que es similar a ADS pero guarda un resultado de 30 bits. No existen equivalentes de TS, SU o AD para 30 bits, pero sí existe un equivalente de XCH (DXCH) que intercambia el valor de A y L en dos palabras de memoria.

Cada acceso extra a una dirección de memoria representa un ciclo extra de 12 µs, por eso una instrucción de suma se ejecuta en 2 ciclos, uno para cargar la instrucción y otro para cargar el valor a sumar al registro A. En el caso de tratarse de una instrucción de suma de una palabra de doble precisión, se debe sumar un ciclo extra más (otros 12 µs), porque se acceden a dos valores consecutivos en memoria para la operación con 30 bits. El acceso a A es directo dentro del procesador, tanto para lectura como para escritura, porque está implementado con lógica digital y no usando núcleos magnéticos, y no requiere de un ciclo extra de 12 µs. A y L son los únicos registros especiales que se pueden usar de forma directa en operaciones matemáticas en este procesador. En otros sistemas se cuenta con mayor cantidad de registros, por ejemplo el 8086 tiene AX, BX, CX y DX y se pueden realizar operaciones entre ellos sin acceder a la memoria. En este caso todos los demás valores se deben guardar en la memoria, y cada acceso a la memoria es un ciclo extra de 12 µs.

El parámetro en una instrucción siempre es una referencia a una dirección de memoria y no existe, a diferencia de la mayoría de los procesadores, instrucciones en *modo inmediato* donde la instrucción incluye un valor para usar en la operación. Por ejemplo, si se quisiera sumar 9 a A, una instrucción de suma en modo inmediato incluiría un 9, pero aquí la única forma de realízalo es incluyendo en la instrucción una dirección que apunta a una parte de la memoria (RAM o ROM, es lo mismo) en donde hay un valor 9.

Realizar una multiplicación con números binarios es bastante simple: al multiplicar X por Y (dos valores de 15 bits) se deben realizar hasta 15 sumas, primero sumando X, luego $X * 2^1$, luego $X * 2^2$, hasta sumar $X * 2^{14}$. Cada multiplicación de X por la siguiente potencia de 2 consiste en, tan sólo, girar X un bit a la izquierda y, en total, X se debe girar 15 veces y a la vez se va sumando al resultado solamente en los casos en que el bit de Y correspondiente al giro sea 1. Por ejemplo 6 * 10 en binario es $0110_2 * 1010_2$ ($2^3 + 2^1$) y para obtener el resultado se debe sumar $01100_2$ (multiplicado * $2^1$) + $0110000_2$ (multiplicado * $2^3$) = $0111100_2$ que en decimal es 60. Si el procesador no dispone de una instrucción para multiplicar es sencillo realizar un programa que multiplique, por ejemplo, por 10: simplemente se suma el valor girado 1 vez, más ese mismo valor girado otras 2 veces. El problema es cuando se deben multiplicar dos valores arbitrarios, porque aquí habrá que implementar un programa que gire un número, si es de 15 bits, 15 veces y según el valor del bit correspondiente del otro valor se va sumando o no. Esta operación ocupa muchas decenas de instrucciones y es extremadamente lenta. Los procesadores más simples no suelen contar con una instrucción de multiplicación y se deja a cargo del programador implementarlas en software, por ejemplo en el Z80, el 6502 o los microcontroladores PIC

16fxx. En esta computadora existe una instrucción de multiplicación (MP) que multiplica un valor de 15 bits en A por otro valor de 15 bits en memoria y guarda el resultado de 30 bits en A (la parte alta) y L (la parte baja) y, al estar implementada en hardware, realiza una multiplicación en apenas 3 ciclos de memoria (35 µs).

La división es una operación similar a la multiplicación: si se divide X / Y se va girando Y, pero a la derecha, comparando con X en cada paso y si es menor o igual se suma $2^n$ (según el paso en que se esté) al resultado y se resta Y de X. Es un poco más complicada la implementación porque primero se debe ajustar el valor de Y hasta girarlo todo lo que se pueda a la izquierda sin perder bits, y en cada paso se debe realizar una resta, primero para comparar y luego, según el resultado se debe guardar o descartar el resultado de esa resta. En esta computadora existe la instrucción de división que, en 6 ciclos de memoria (70 µs), divide un valor de 30 bits (en A y L) por un valor de 15 bits en memoria y el resultado (de 15 bits) se almacena en A y el resto (o módulo) queda en L. Esta instrucción no hace ninguna comprobación de los parámetros para ver que el divisor no sea 0 o que el resultado quepa en 15 bits y si no se cumplen estos requisitos el resultado no tendrá ningún sentido, por eso el programador debe usarla con cuidado. En otras arquitecturas que incluyen una instrucción de división, por ejemplo en el 8086, el procesador dispara una excepción cuando los valores de entrada no pueden usarse en la operación.

**Instrucciones de salto y pila de llamadas**

No es muy útil un programa que se ejecute siempre avanzando en una sola dirección con una instrucción atrás de otra, por eso es necesario cada tanto cambiar el flujo del programa y hacer que se continúe la ejecución del programa en otra parte. Para esto existen instrucciones que modifican el valor del registro Z, el que guarda la dirección en ejecución. Las instrucciones básicas son TC (Transfer Control) y TCF (Transfer Control to Fixed) y ambas lo que realizan es cargar en Z una dirección de memoria para que el programa continúe ejecutándose ahí. TCF sólo permite saltar a una región en memoria ROM en cambio TC puede saltar a cualquier dirección pero, además de eso, tiene una funcionalidad extra utilizando otro registro especial de la computadora: Q.

Una estructura presente en la mayoría de los procesadores es la pila de llamadas que guarda la dirección en donde debe continuar el programa cuando termine de ejecutar una función. La forma en que se usa es así: el programa llama a una función con una instrucción especial (CALL en el 8086) y esta guarda en la pila la dirección a donde debe volver y, al concluir la ejecución de la función, se llama a la instrucción para terminarla (RET en el 8086) y esta toma la dirección de la pila para regresar a donde estaba antes, que es la instrucción siguiente al CALL. Esto proporciona mucha flexibilidad al programador, porque con solamente poner una llamada a una función sabe que cuando termine de ejecutarse el programa continuará donde estaba. Lo mismo se simplifica a la hora de escribir la función, que se escribe sin importar desde donde se ejecutará porque se sabe que automáticamente cuando finalice su ejecución volverá a donde la llamaron. Lo útil de que las direcciones de

retorno se guarden en una pila es que permite guardar varias direcciones, así una función puede llamar a otra función, y esta última a otra más, y a medida que se vayan terminando la ejecución cada una volverá a donde debía.

Esta computadora, para simplificar el hardware, no cuenta con una pila: no existe una instrucción para llamar a una subrutina ni para terminarla. La instrucción TC guarda en el registro Q la dirección de retorno (la siguiente a la del salto) pero es por software que se debe implementar los mecanismos para regresar a la dirección Q una vez terminada la ejecución: algo así como una pila de llamadas pero de un único nivel. Como es un solo registro, si una función quiere llamar a otra, debe guardar el valor actual de Q para no perderlo y recuperarlo luego, y todo en direcciones de memoria propias de la función esa.

**Saltos condicionales**

El programa necesita, en determinados puntos, optar entre más de una opción según si se cumple una condición o no, y para eso se usan las instrucciones de salto condicional que hacen que cuando se ejecuten el programa continúe con la siguiente instrucción o siga en otra parte, dependiendo de un determinado criterio. Esto se usa, por ejemplo, para hacer que una acción se ejecute N veces utilizando una variable que se incrementa cada vez que se realiza la acción y salta al inicio de la operación mientras la variable no tenga el valor N. También para tomar decisiones en base periféricos, por ejemplo leyendo el estado de un botón se salta a una parte del programa que actúa cuando ese botón está presionado. Lo más común en distintas arquitecturas es la existencia de un registro especial de flags o banderas, donde se guardan distintos estados de la última operación: si el resultado fue 0, si fue positivo, negativo o hubo un desborde. Existen distintas instrucciones que realizan un salto en base a una de esas condiciones y en el 8086 existen muchas instrucciones de salto con varias de estas combinaciones: si el resultado fue 0, si fue distinto de 0, si hubo desborde, si no hubo desborde, si hubo desborde o es 0, etc. También suele existir una instrucción de comparación (normalmente llamada CMP, por compare) que lo que realiza es una resta entre los dos valores pero el resultado de esta resta no se almacena en ningún lado, sino que sólo se actualizan los flags: por ejemplo, si el resultado fue cero (significa que los dos valores eran iguales) o si hubo desborde o no (sirve para saber cuál valor era el mayor de los dos). Lo útil de esta instrucción de comparación es que, como no modifica ninguno de los valores, estos se pueden seguir utilizando para otra cosa.

En esta computadora los saltos condicionales con más simples y limitados: existe la instrucción BZF que salta a una dirección si el valor en A es 0 y otra instrucción, BZMF, que salta si A es 0 o negativo. Ambas instrucciones trabajan con el valor actual de A y no hay una instrucción de comparación: para hacer una comparación de dos valores hay que cargar el valor en A, hacer una resta con el otro valor a comparar y saltar en base a la respuesta, pero el valor en A se perdió luego de la resta y si se lo quiere comparar con otra cosa hay que volverlo a cargar.

Existe una instrucción que, entre otras cosas, realiza una comparación pero es muy compleja y se llama CCS (Count Compare and Skip) y lo que realiza es copiar el valor de una variable de RAM en A restándole 1. Luego puede realizar una de 4 posibles acciones dependiendo del valor original de la variable: si era mayor a 0 continua con la ejecución de la siguiente instrucción, si es 0 se saltea una instrucción, si es negativo se saltea 2 instrucciones y si es el 0 negativo (existente en complemento a 1) saltea 3 instrucciones. Esta instrucción puede ir acompañada de 4 saltos que se dirigirán a distintas secciones del programa según el valor leído. El decremento que realiza del valor original en A sirve para implementar un loop de forma más o menos simple.

No hay un registro de flags con el resultado de la última operación, pero existe algo que, en algunos casos, cumple una función similar. Desde el punto de vista del programador esta máquina es de 15 bits, pero la electrónica es de 16 bits: las memorias y los buses tienen ese tamaño, a pesar de que sólo 15 bits tienen información, el bit extra es redundante, su valor depende del contenido de los otros bits y la electrónica verifica que esto se cumpla. Este bit extra en los registros A, L y Q, como no necesitan control de paridad, se aprovecha para otra función: detección de desborde. Cuando se realiza una suma o resta sobre A pero el resultado se va de rango, ya sea por arriba o por abajo, se indica en este bit de desborde. Si se carga un valor en A desde la memoria se resetea este indicador de desborde pero si se copia el contenido de A a L o a Q, o en sentido inverso, se copia también este estado de desborde junto con los 15 bits normales del valor. A diferencia de un registro de flags, que es global, este bit de estado está pegado a cada registro del procesador. Con este indicador de desborde se puede realizar una serie de operaciones matemáticas sobre A y recién al final revisar si hubo desborde en alguno de todos los pasos ejecutados, a diferencia de un flag de desborde que únicamente indica si ocurrió o no en la última operación.

Se pueden tomar decisiones en base al estado del desborde de A, por ejemplo, la instrucción OVSK saltea una instrucción si A está en desborde. TS, que guarda el valor de A en una dirección de memoria, se comporta de manera distinta cuando A está en desborde: guarda el valor de A en memoria pero luego reemplaza el valor de A por 1 o -1, según el signo de A y se saltea la siguiente instrucción. Este comportamiento tan extraño se usa para sumar números formados por más de una palabra así luego de sumar un número que desbordó se reemplaza A por el valor que hay que sumarle al siguiente y si no se ejecuta una instrucción para pasar A a 0 que se saltea en caso de desborde. Esto se realiza en otros procesadores de forma más simple por medio del flag de acarreo que participa en la suma de cada número sumándose al resultado y luego guarda el estado de desborde de la última suma realizada. Cuando se suma un número que no depende de una suma anterior se pone a 0 el acarreo o, si existe, se usa una instrucción de suma que lo ignora.

**Manejo de índices**

Una funcionalidad que da mucho poder al programador son las instrucciones indexadas: en un modo indexado una instrucción accede a una dirección de memoria que no

está fija en la instrucción sino que depende de una variable. Por ejemplo, en vez de acceder a la dirección 456 se accede a 456 + X, donde el valor de X varía en el tiempo. En la mayoría de arquitecturas de computadoras X es un registro especial que permite este tipo de accesos. Por ejemplo en el 6502 se puede leer una dirección de memoria indexada por el registro X (LDA 456,X ) y en el 8086 por el registro BX (MOV AX,[456+BX]).

En esta computadora se aprovecha que el parámetro de la instrucción es una dirección de memoria entonces existe una instrucción especial llamada INDEX que carga un valor de 12 bits en un registro interno que se le sumará al parámetro de la próxima instrucción. Cada instrucción siempre deja en 0 este registro exceptuando INDEX que le carga un valor y luego, cuando se ejecuta la siguiente instrucción, al parámetro de la instrucción le agrega este valor.

Por ejemplo si se hace:

```
INDEX 333 (en la dirección 333 hay una variable que indica que valor
          leer)
CA     456 (lee en A el valor que está en la dirección 456)
```

En este ejemplo en 456 se tiene una serie de valores y en la dirección 333 hay una variable que indica cual de esos valores (de 0 en adelante) hay que leer en el registro A.

Lo poderoso de esta forma de implementar la indexación es que la misma instrucción INDEX sirve para indexar cualquier tipo de instrucción, por ejemplo una de manejo de memoria, o una aritmética o de salto. En otras arquitecturas los modos indexados son limitados y no se pueden realizar cualquier tipo de combinación en cualquier instrucción.

**Instrucciones de Entradas y Salidas: Canales**

Las instrucciones suelen trabajar sobre la memoria, tanto RAM como ROM, que el procesador ve como una sola cosa dentro de las 4096 direcciones que puede manejar por vez. En otros casos se necesita trabajar sobre periféricos, leyendo el estado en que se encuentra una señal de entrada para saber, por ejemplo, si un botón está presionado, o para establecer el estado de una señal de salida, por ejemplo para encender o apagar un motor. En esta computadora para comunicarse con los periféricos se cuenta con un área de direccionamiento independiente de las 4096 direcciones de memoria. Para acceder a este espacio de direcciones de entradas y salidas no se pueden usar las instrucciones normales sino instrucciones especiales dedicadas a esto.

La existencia de un área de direccionamiento de entradas / salidas distinto del espacio de direccionamiento de memoria es común en varias arquitecturas de microprocesadores, como la línea Intel 80x86 que se utiliza en las PCs y en el Z80 y sus derivados. En otras arquitecturas se opta por manejar las entradas / salidas como direcciones de memoria y que la electrónica externa se encargue de su interfaz, evitando así tener que contar un conjunto de instrucciones distinto para el manejo de periféricos. En la

línea de microprocesadores ARM, los microprocesadores Motorola y los PICs se utiliza este diseño de periféricos mapeados en memoria.

Las instrucciones de manejo de entradas / salida se manejan por medio de instrucciones especiales que, en lugar de tomar el valor del parámetro como una dirección de memoria, lo toman como un canal de entradas / salidas, pero en vez de 12 bits utiliza sólo 9 para identificar uno de los posibles 512 canales. Hay canales de que son de salida y tienen un buffer que guardan el último valor que la computadora les escribió y lo envían hacia periféricos y hay otros que son de entrada y solamente se puede leer su estado. La distribución y configuración de cada canal no depende del software sino que ya viene fijo en el hardware y su diseño está adaptado al uso que se le dará. En el formato normal de instrucciones existe una sola operación de manejo de periféricos, pero como solamente se usan 9 bits para indicar el canal se utilizan 3 bits extra para indicar que operación realizar sobre el canal, por eso se termina contado con 7 instrucciones de este tipo y las principales son READ, que copia en A el estado de los 15 bits de un canal de entrada y WRITE copia el contenido de A en un canal de salida (Tabla 2).

| Operación | Sub-Operación | Canal |
|---|---|---|
| 14 13 12 | 11 10 9 | 8 7 6 5 4 3 2 1 0 |

**Tabla 2.** Formato de una instrucción de entradas / salidas

Una particularidad de este diseño es que existen operaciones para manejar canales de periféricos que no están disponibles cuando se trabaja sobre la memoria y son operaciones de manejo de bits: OR y XOR (OR exclusivo). Estas instrucciones de manipulación de bits resultan muy útiles en los canales porque permiten manipular bits de forma independiente dentro de un mismo canal y de esa forma se pueden usar los 15 bits de información de un canal para distintos periféricos y manipular cada uno por separado. Por ejemplo, en un canal con un bit se puede activar un motor y con otro se activa otro, pero cada uno se controla de forma independiente.

**Interrupciones**

Una característica presente en la mayoría de los procesadores son las interrupciones, que hacen que cuando se recibe un evento externo, el procesador salte automáticamente a una rutina que realiza una acción pero, antes de ejecutar esta rutina, se guarda el estado del procesador para que, una vez que se atendió a la interrupción, se continúe ejecutando el programa original como si nada hubiera ocurrido. Las interrupciones están pensadas para que el procesador se ocupe de eventos que requieren una pronta atención, pero sin entorpecer al programa en ejecución ni obligarle a chequear constantemente si ocurrió un evento. Por ejemplo, si se desea atender dentro de unos milisegundos un evento que ocurre mientras se está realizando un cálculo que puede llevar 50 ms, lo que se puede hacer es que,

mientras se realiza el cálculo, el programa constantemente verifique si se produjo el evento y lo atienda, pero esto desperdicia muchos recursos, en instrucciones de programa y tiempo de ejecución. Por medio de una interrupción se simplifica el programa porque mientras se realiza la lenta operación matemática, este software sólo se dedica a eso y cuando ocurre el evento se ejecuta la interrupción inmediatamente y luego se continúa con el cálculo como si nada hubiera ocurrido.

Esta computadora cuenta con 11 señales de interrupción y cada una de ellas tiene una dirección de ROM en donde se encuentra la rutina que la procesa. Si existiera una pila de llamadas se guardaría allí la dirección de regreso, pero aquí se utiliza una palabra en RAM para esa función.

Las interrupciones se pueden habilitar y deshabilitar para evitar que entre una interrupción y retrase la ejecución mientras el programa está realizando una operación que se tiene que realizar muy velozmente, pero es conveniente no tenerlas deshabilitadas mucho tiempo porque si no el procesador no podrá atenderlas con la rapidez necesaria. Mientras se ejecuta una interrupción automáticamente se deshabilitan las interrupciones para que no entre otra antes de terminar el procesamiento de la anterior.

**Direcciones de registros especiales**

Las primeras direcciones de memoria no apuntan ni a la RAM ni a la ROM sino que a través de ellas se accede a los registros internos del procesador y a otros registros especiales y de esta forma se pueden utilizar las mismas instrucciones que se usan con variables en memoria de RAM para manipular estos registros sin necesidad de instrucciones especiales. El acumulador (A) está en la dirección 0, y entonces, por ejemplo, se puede ejecutar la instrucción AD, para sumar un valor a A, y sumarle el valor en la dirección 0 y de esta forma se suma A sobre A, o sea que equivale a multiplicar por 2. El registro L viene a continuación de A de la misma forma en qué se almacenan los valores de doble precisión y esto permite utilizar las instrucciones que manipulan de a 30 bits para modificar A y L a la vez. El siguiente registro, en la dirección 2, es Q, que guarda la dirección de retorno de la instrucción TC.

El registro Z, el que guarda la dirección de la próxima instrucción a ejecutar, está en la dirección 5 y esto permite que se pueda modificar el flujo de ejecución cargando un valor en esta dirección en vez de usar una instrucción de salto, lo que da más flexibilidad en algunos casos. Junto a Z existen 3 registros del mapeador de memoria: EBANK (Eraseable Bank, banco de memoria borrable) en la dirección 3, FBANK (Fixed Bank, banco de memoria fija) en 4 y BB (Both Banks, ambos bancos) en la dirección 6 que es una mezcla de los dos anteriores y apunta a los mismos bits pero agrupados de otra forma. La forma en que están organizados estos registros permiten modificar Z y FBANK a la vez utilizando una instrucción para guardar un valor de doble precisión en memoria (DXCH), y así se puede saltar a otra dirección al mismo tiempo que se modifica la configuración del mapeador, lo que significa que se puede realizar un salto largo a un área del programa no visible al momento del salto.

En la dirección 7 existe siempre, de forma fija, un valor 0 y, como este valor no es modificable, se puede usar con una instrucción de intercambio (LXCH o QXCH) para intercambiar el contenido de un registro con esa dirección y lo único que realizará es cargar el valor 0 en el registro con una sola instrucción. Puede parecer que lo más lógico hubiera sido guardar un 0 en alguna posición de ROM pero entonces esa dirección no se podría utilizar en instrucciones que solamente existen en el área de RAM (como LXCH o QXCH). En las direcciones 8 a 15 se encuentran registros especiales usados durante la interrupción, uno que guarda en qué dirección continuar la ejecución luego de terminar la interrupción y también otros registros especiales también usados para guardar el estado del procesador antes de la interrupción que ayudan a que la ejecución de la interrupción no altere el normal flujo del programa. La particularidad de estas direcciones es que guardan los 16 bits de los registros sin usar el bit extra como paridad sino que permiten guardar el indicador de desborde para que no se pierda en la interrupción.

Luego existen 4 direcciones que implementan, de forma externa al procesador, algunas operaciones de giros de palabras para las que no existen instrucciones. Este tipo de operaciones es muy útil cuando se quiere manipular valores por bits, sobre todo cuando se debe ahorrar memoria y se guardan distintos valores dentro de una misma palabra en diferentes bits. Cuando se guarda un valor en una de estas direcciones, este se almacena modificado: una dirección lo gira para un lado, otra para el otro y según cuál de las cuatro se elija se selecciona si se pierde un bit en el giro o vuelve a entrar en la posición opuesta. Esta implementación ahorra complejidad al procesador, es casi como un coprocesador muy básico, aunque tiene el inconveniente de que es global, accesible a todos los programas e interrupciones por igual, y es necesario administrar como se los usa porque se pueden producir conflictos si dos programas los usan a la vez o una interrupción modifica su contenido.

Todas estas direcciones, a pesar de estar implementadas de forma directa dentro de la computadora y de no necesitar todos los pasos para acceder a la ROM de sogas o a la RAM de núcleos magnéticos, se usan a la misma velocidad, necesitando un ciclo completo de 12 µs para leer o escribir una palabra.

Además de estas direcciones de memoria, en los canales de entradas / salidas se realiza algo similar: se puede acceder a los registros L y Q como en canales dedicados y esto permite utilizar en estos registros las operaciones de manejos de bits solamente disponibles en las instrucciones de canales, como el OR y el XOR.

**Contadores**

La computadora dispone de registros contadores que se incrementan con eventos externos. Es común, sobre todo en microcontroladores, la existencia de circuitos que cuentan pulsos producidos externamente. Por ejemplo si se tiene un sensor que genera un pulso cada vez que una persona pasa por un pasillo y se conecta a la entrada del contador, el programa con leer ese contador cada tanto tiempo puede saber cuánta gente pasó por el

pasillo sin tener que estar atento constantemente a esa tarea. Esta computadora tiene un mecanismo similar, pero con una diferencia muy importante: el circuito del contador no es externo sino que depende el procesador. Para ahorrar lógica digital necesaria para realizar el incremento del valor y para conectar el contador al bus del procesador, lo que se implementó es un sistema en que el procesador recibe un pulso y deja lo que estaba haciendo y realiza una suma en un contador en particular en la memoria RAM. De esta forma se reaprovecha la lógica digital dedicada a la aritmética propia del procesador y el manejo de la memoria. Es un mecanismo similar a una interrupción con la diferencia de que las interrupciones están pensadas para usos muy generales y deben ejecutar un programa que guarda el contenido actual de lo que modifica (por ejemplo el registro A), realiza su tarea y luego le indica al procesador que terminó y debe continuar con el programa normal. Toda esta secuencia del manejo de interrupciones consume mucho tiempo de ejecución y los contadores están pensados para recibir muchos eventos. En este caso, al llegar el evento lo que ocurre es que se ejecuta una sola operación de incremento a una dirección de RAM en particular. También cada contador tiene una señal para decrementar ese valor y así con estas dos señales se puede controlar el uso de ese contador en particular.

La principal contra de esta técnica es que cada pulso que entra le quita tiempo de procesamiento a la computadora, que, si bien no es tanto el tiempo que le quita comparado con una interrupción, se pierden varios ciclos que el programa no puede usar mientras se actualiza un contador.

Existe una serie de contadores y cada uno se usa de forma distinta, algunos disponen de incremento y decremento y otros únicamente cuentan en un sentido, algunos disparan una interrupción cuando se desbordan, o sea cuando llegan al valor máximo y ya no pueden almacenar un número mayor. La lógica de cada contador está ajustada en hardware para el uso específico que se le dará en el programa. Hay contadores que trabajan con complemento a 2 en vez de complemento a 1 y existe uno que se utiliza como una UART para convertir una entrada serial en una palabra dentro de la computadora y que funciona también con dos líneas: cuando entra un pulso en una se gira al contador y se agrega un 1 y cuando el pulso viene en la otra se gira igual pero se agrega un 0. Con 15 pulsos repartidos entre las dos líneas se tiene una palabra completa y al llegar el pulso 16 se dispara una interrupción indicando que entro una palabra por la entrada serial.

**Timers**

Las computadoras funcionan con una señal de reloj que sincroniza los pasos necesarios para cargar descodificar y ejecutar las instrucciones en todas sus etapas, pero aparte de este reloj es necesario que el programa tenga referencias de tiempo externas para, por ejemplo, realizar una tarea que tiene que ejecutar cada 2 segundos y saber exactamente cuándo hacer esto, o cronometrar procesos físicos, como el tiempo que le lleva a un motor encenderse. También es necesario que el programa conozca la hora actual para iniciar una maniobra en el momento exacto que se espera, por ejemplo si la tripulación le

indica realizar una maniobra a las T+4:32:16 horas (4 horas 32 minutos y 16 segundos después del lanzamiento en el horario que se usa en la misión) el programa de la computadora espera hasta que llega ese instante para iniciar la maniobra, y la hora a la que accede el programa debe ser la misma con la que cuenta la tripulación para que todo funcione correctamente.

Para que el programa tenga referencias de tiempo externa se cuenta con 5 timers [7], cada uno con usos distintos. Los timers están implementados como contadores que solamente incrementan y el programa lee el valor en memoria de cada contador y sabe cuántos pulsos transcurrieron desde un momento dado según el timer. De las señales que recibe cada contador, 4 de ellas vienen con un periodo de 10 ms, por eso este es el lapso mínimo de tiempo que pueden medir, y el quinto tiene un periodo de 0,645 ms y se lo utiliza para cronometrar de forma más precisa.

El primer timer se usa para medir el tiempo de la misión (tiempo desde el lanzamiento) y cuando este timer se desborda, en lugar de generar una interrupción, incrementa al siguiente contador que se convierte en los 15 bits superiores del timer, así se cuenta en cada momento con un timer de 30 bits midiendo el tiempo transcurrido desde el lanzamiento con una precisión de 10 ms.

Los otros 4 timers se utilizan para cronometrar distintos procesos y periódicamente el programa modifica su valor, para aprovechar la propiedad de cada contador de generar una interrupción cuando se desborda. De esta forma si se desea que se produzca una interrupción dentro de 200 ms, se le pone a un timer el valor 16383-20 para que luego de 20 pulsos de 10 ms se desborde el contador y produzca una interrupción y luego en la interrupción se realiza la tarea planeada.

**Watchdog**

Una característica novedosa de la computadora, compartida con los modernos microcontroladores, es la presencia de un timer Watchdog (perro guardián) que en esta computadora se llamaba night-watchman (vigía nocturno) [3]. Este timer decrementa un contador constantemente, y cuando el valor del contador llega a 0 luego de 640 ms, lo que hace es resetear la computadora.

Para que el programa pueda ejecutarse correctamente es necesario que el mismo programa periódicamente actualice el contador del night-watchman para que nunca llegue a 0. Esta técnica sirve para detectar errores de software graves, por ejemplo cuando la computadora no está ejecutando el programa correcto, sino que quedó trabada ejecutando sin fin algo que no es válido. Con esto se minimizan las posibilidades de que un programa descontrolado realice operaciones peligrosas o al menos se evita que lo realice durante un tiempo largo.

**Entradas analógicas**

Cuando una computadora debe obtener datos del mundo exterior, lo más directo es leer entradas lógicas con dos estados, por ejemplo un botón que está presionado o suelto, la computadora lee este valor como un 1 o un 0 según el estado del botón.

Es más complejo cuando se debe obtener el estado de un componente que tiene más de dos estados, por ejemplo la inclinación de la palanca de mando o la medición de un acelerómetro. Si tomamos cada uno de esos posibles valores como un rango de voltajes, por ejemplo cuando la palanca en el eje X está toda hacia la izquierda, la señal que genera es de 0 V, y cuando está en el otro extremo, toda hacia la derecha la señal es de 5 V, debemos codificar este rango de voltajes como números digitales que la computadora pueda procesar. Aquí se debe seleccionar el rango de voltajes y el rango de números, por ejemplo se puede hacer que 0 V se represente como el número 0 dentro de la computadora, y los 5 V, el máximo valor de voltaje se represente como 255, en cuyo caso se tendría una conversión a 8 bits. Si en cambio utilizamos 14 bits, los 5 V se representarían como el número 16383, el máximo valor con 14 bits. El circuito que realiza la conversión de un valor analógico a un número digital que la computadora puede leer es el conversor analógico-digital, conocido normalmente por la sigla AD.

La mayoría de los microcontroladores poseen una o varias entradas AD para poder interactuar con el mundo exterior y también existen circuitos dedicados a esta función que se pueden conectar a una computadora. La forma electrónica de convertir una cantidad continua en un valor digital es con comparadores, que comparan el valor analógico con distintos valores ya conocidos. Para convertir una señal analógica a una señal digital de 8 bits, no hacen falta 8 comparadores, sino 256 ($2^8$) y luego se debe ver cuál es el mayor comparador que dio una señal menor para saber cuál es el valor digital que más se le acerca. Si se sacrifica velocidad a la conversión se puede hacer lo mismo con un solo comparador pero comparando secuencialmente con distintos valores.

En la computadora de las naves Apolo no había una forma práctica de implementar una conversión AD electrónica para cada señal necesaria porque requería muchos circuitos usando la tecnología de la época. A lo que se recurrió es a una conversión electromecánica que utilizaba los contadores de la computadora. De forma similar a un mouse a bolita, donde el movimiento de la bola se transmite a unos rodillos que generan un tren de pulsos que va a la computadora y luego esta los cuenta y de ahí obtiene cuanto se desplazó la bolita (y el mouse) para mantener actualizada las coordenadas. Aquí un sistema electromecánico mide el estado de un sensor y cuando cambia el valor envía una serie de pulsos de incremento o decremento según en la dirección en que se movió. La computadora, al recibir esos pulsos, actualiza el valor del contador y como aplica los mismos cambios obtiene el valor actual que tiene el instrumento físico.

Al utilizar contadores para esta función se tiene el problema de que se roba tiempo de ejecución al procesador. Si el conversor tira una serie de pulsos muy grande le quitará mucho tiempo al programa. Si un sensor quedara oscilando entre un rango de valores

enviando todo el tiempo pulsos de incremento y decremento, el procesador perdería capacidad de procesamiento al recibir tantos eventos, y un problema de este tipo ocurrió cuando se estaba realizando el primer alunizaje en la misión Apolo 11.

**Mapeo de memoria**

En un mapeo de memoria externo, un circuito ajeno al procesador le agrega más bits a la dirección entregada por este para acceder a la memoria completa. Como el procesador solamente es capaz de utilizar direcciones de 12 bits, este circuito externo le debe agregar los 4 bits extra necesarios para tener un tamaño de dirección que pueda apuntar a cualquiera de las direcciones existentes en el hardware. En este caso esto se hace dividiendo a la memoria en distintos rangos y aplicando lógicas distintas para cada uno.

La RAM está mapeada en las primeras 1024 direcciones, aunque las primeras son en realidad registros internos del procesador, pero se considera todo este rango como RAM. En las direcciones de 0 a 767 se accede directamente a la primera parte de la RAM, y las siguientes 256 direcciones miran una ventana de 256 palabras seleccionables de las 2048 disponibles. La selección de uno de los 8 posibles bloques de 256 palabras se hace escribiendo al registro especial EBANK. De esta forma hay un área de memoria RAM común que se puede acceder en cualquier momento y el resto de la memoria RAM se debe manejar de a bloques de 256 palabras.

En el caso de la ROM se utilizan las 3072 direcciones siguientes que puede indicar el procesador. En las primeras 1024 direcciones de este rango se ve una ventana de 1024 palabras de una selección de uno de 32 bloques en que se divide la ROM física que se seleccionan con el registro especial FBANK. Las últimas 2048 direcciones apuntan siempre a la última parte de la memoria ROM. El problema que surgió en la ROM es que el esquema este permite manejar hasta 32 kB de palabras de esa memoria y el diseño fue hecho cuando el estimado eran de 24 kB de ROM, pero con el tiempo el programa creció hasta ocupar 36 kB. Para poder acceder a esos 4 kB no visibles sin hacer modificaciones importantes a la arquitectura de la computadora, se le agregó un bit llamado SuperBank , que se maneja en un canal de salida, no como un registro, y este bit se le concatena a la dirección formada por FBANK en algunos casos, lo que permite seleccionar más de 32 bloques de 1 kB en la ventana.

**FABRICACIÓN**

En el MIT se diseñaba la computadora y se probaban los diseños primero simulándolos por software en computadoras más grandes y luego armando prototipos cableados a mano que ocupaban varios armarios enteros, pero lo que no realizaban era la fabricación de los equipos que viajarían al espacio. De esto se encargaba Raytheon [13], una empresa de larga trayectoria en tecnología aeroespacial y militar, creadora de una gran cantidad de sistemas de misiles y también del horno a microondas, y al igual que el MIT,

residente en Massachusetts. Era en Raytheon que estaba el equipo de costureras dedicado a armar los seis módulos de memoria de soga que usaba cada computadora pero aquí también se encargaban de la fabricación de toda la electrónica.

La lógica de la computadora estaba formada por 24 módulos [3], cada uno compuesto por dos circuitos impresos donde había, en cada uno, dos filas de 30 integrados todos iguales y alineados en la misma dirección. Los circuitos impresos se crearon en la década del 1950 y en poco tiempo se popularizaron entre los fabricantes de artículos electrónicos, que en aquella época eran principalmente radios y televisores. Anteriormente usaban los componentes electrónicos soldados en armazones de cables y toda la red de conexiones era armada manualmente al momento de ubicar los componentes en su lugar en cada equipo. Con los circuitos impresos se simplificaba el armado porque en una plaqueta fabricada masivamente ya vienen todas las uniones eléctricas y sólo hay que ubicar cada elemento en un lugar pre asignado. La forma en que se aprovecha el espacio es más eficiente en un circuito impreso que en uno cableando manualmente y la reducción en tamaño es mayor mientras más capas conductoras se utilicen: si se usa una sola, en un lado de la placa, cuando una conexión tenga que cruzarse con otra habrá que usar un cable del otro lado para cruzar la pista conductora. En cambio, si se usan dos capas conductoras, una a cada lado de la placa, en el propio circuito impreso se realiza el cruce. En aquella época se solían usar circuitos de una sola capa pero en esta computadora los impresos eran de cuatro capas, dos externas y dos internas, lo que permitía mayor flexibilidad para interconectar las cientos de entradas y salidas lógicas que había en los 60 circuitos integrados (cada uno una doble compuerta NOR de tres entradas).

Los chips, antes de ser ubicados en las placas, pasaron un exhaustivo control de calidad para verificar el funcionamiento individual y del lote completo del que partieron. Los fabricantes de los integrados tenían una línea de producción aparte para los componentes que usaría la NASA porque los controles de calidad eran mucho más estrictos que el resto y, como todavía no estaba muy automatizada la producción de circuitos integrados, las empleadas (solían ser mujeres en esta tarea) de la planta productora que debían montar cada chip verificaban con microscopio que el chip no presentara anomalías. La planta productora era visitada periódicamente por astronautas para recordarles a todos los trabajadores lo importante que era que no cometieran errores para poder cumplir los objetivos de la NASA. En Raytheon cada chip era sometido a una prueba en una centrifugadora para verificar su resistencia a aceleraciones de muchas G y luego se los exponía a gas helio a alta presión para después ponerlos en una cámara de vacío, comprobando que no hubiera absorbido nada de ese gas y de esta forma demostrando que el encapsulado no presentara fisuras. Luego se los ponía en zócalos especiales en una máquina que realizaba la prueba de burn-in: los hacía funcionar durante una semana en condiciones de voltaje y temperatura superiores al uso normal y se verificaba que durante todo ese tiempo nunca fallaran. Una vez que se tenían chips debidamente probados, se podían ubicar en los módulos de la computadora. Pero como las técnicas de montaje superficial no estaban muy desarrolladas en aquella época, no se soldaban a la placa con

estaño sino que las patas de los chips se fundían con las pistas del circuito impreso, utilizando presión y una temperatura mucho mayor a la necesaria usando estaño. El módulo armado tenía en un extremo un conector de unos 20 cm de ancho con 280 pines en cuatro hileras donde se exponían distintas entradas y salidas del circuito lógico formado por todas las compuertas y sus interconexiones.

Las demás placas electrónicas de la computadora estaban armadas con la técnica conocida como cordwood (atado de leña) que consistía en utilizar dos circuitos impresos con los componentes puestos en el medio con una pata soldada a cada plaqueta. Mucha de la electrónica de las naves Apolo se armaba con esta misma técnica porque de esta forma se lograba reducir mucho el tamaño de la electrónica aunque era muy difícil reparar un circuito así. Además de los módulos lógicos con los circuitos integrados, estaban los de memorias ROM y RAM, los de regulación de tensión, generación de pulsos y controladores de memorias, entre otros. Todos los módulos, ya sea los lógicos, los de memoria y los de otras funciones, se ponían en cartuchos rectangulares que tenían un conector con muchos pines de uno de los lados más anchos y eran probados por separado en computadoras grandes con adaptadores especiales y programas que verificaban que respondieran como era debido. Por ejemplo, una memoria ROM era leída y su contenido comparado con la información que debía tener y de los módulos de lógica se probaban las distintas combinaciones posibles para ver que todo funcionara correctamente. Para hacer andar la computadora se debían unir todos estos módulos entre sí y para eso se insertaban dos bandejas que contenían en total tres hileras de sockets para los cartuchos. Estas bandejas no tenían los sockets en un circuito impreso porque era muy complejo realizar así todas las interconexiones entre los miles de pines de todos los módulos, en una computadora con buses de 15 bits que requiere toda esta cantidad de señales uniendo múltiples módulos además de muchas otras conexiones. Se usaban entonces cables para interconectar los pines porque así existía mayor libertad, al no estar limitados a trabajar sobre unos pocos planos, como en un circuito impreso [13].

La técnica de cableado y unión utilizada en cada bandeja era wire-wrap: estos son unos cables muy finitos (útiles para transportar señales y no corrientes importantes), maleables y con una capa de aislamiento muy frágil que es fácil de sacar con un poco de fuerza. En la actualidad es común utilizar estos cables para realizar prototipos, por lo práctico del cable y la facilidad para remover la aislación con el soldador, pero la técnica de Wire-wrap es más compleja: una pinza toma el cable y lo enrolla con varias vueltas alrededor de un pin que tiene un terminador en el extremo para que el cable no escape, y al ajustarlo el aislamiento se rompe junto al pin y quedan unidos por la presión ejercida, sin necesidad de soldar. En un espacio relativamente tan pequeño como ese, no era posible cablearlo a mano sin una planificación detallada, porque una acumulación excesiva de cables en una zona podría entorpecer el cableado del resto y además las posibilidades de cometer errores eran gigantescas. Por eso se utilizaba una máquina de ruteo de wire-wrap controlada por una computadora que tenía cargada en tarjetas perforadas la disposición de los cables, que previamente había sido calculada en otra computadora para utilizar un conexionado óptimo en que hubiera poco amontonamiento de cables [13]. Una vez que este cableado estuviera

completo y verificado, se rellenaba con resina epoxy para que cuando se solidificara mantuviera todas las conexiones fijas y ningún cable se soltara, aunque yo no sería posible ninguna reparación a esta parte de la computadora.

Una vez interconectados los módulos en las dos bandejas apiladas dentro del gabinete, la computadora se probaba exhaustivamente, con pruebas de vibración y temperatura para descartar cualquier problema de conexiones poco firmes. Cuando ya se habían pasado todas las pruebas en forma exitosa, se sellaba la parte del gabinete que contenía la mayoría de los módulos y a partir de ese momento sólo podrían reemplazarse con facilidad los seis módulos de ROM para actualizar el firmware de la computadora. El gabinete que contenía toda la electrónica estaba hecho de magnesio porque así se ahorraban unos 5 kg de peso en relación a un gabinete de aluminio, pero la contra es que el magnesio es reactivo y se debió estudiar la forma de trabajarlo para que no provocara corrosión en ningún componente. De la parte externa están los conectores para obtener 28 V de continua y las 110 señales de entrada y 134 de salida de los periféricos de la nave que serán controlados y monitoreados por la computadora. El equipo en ese sentido es modular porque se pude sacar con relativa facilidad del panel de cualquiera de las dos naves y se puede reemplazar por otro similar con el firmware que le corresponde al tipo de nave (Figura 9).

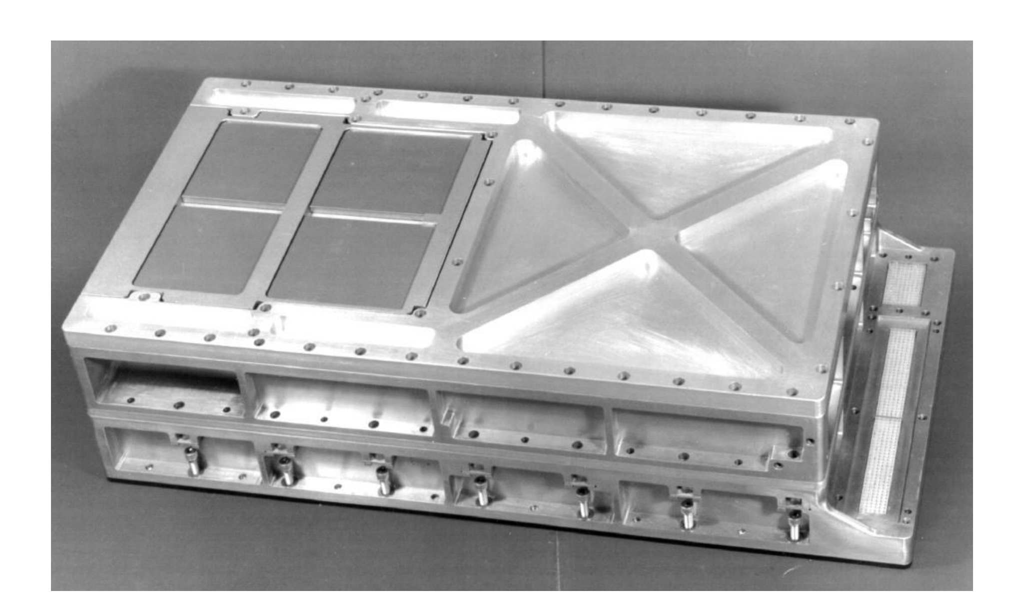


**Figura 9.** Gabinete que contiene todos los módulos electrónicos de la computadora.

## Modelos intermedios de la computadora

Antes de crear el Block II, el modelo final de la computadora que viajaría a la Luna, e incluso antes del Block I, el primer modelo que entraría en producción y se podría usar en el Módulo de Comando, se hicieron varios diseños preliminares. Con anterioridad al inicio del programa Apolo se habían creado varias versiones de la computadora [5]. La primera, el modelo 1A, ya era de 15 bits y tenía el mismo formato de instrucción (3 bits de operación y

12 de dirección) que, como la memoria era muy reducida (4000 palabras de ROM y 128 de RAM) podían apuntar a cualquier parte de la computadora sin necesidad de segmentar las direcciones en distintos bancos de memoria. El diseño era simple y solamente disponía de ocho instrucciones y no había instrucciones extendidas.

Para superar las limitaciones encontradas en la primera versión se inició el proyecto de crear un segundo modelo más complejo y de 24 bits, pero este se abandonó y se pasó al Modelo 3 con las mismas características que el primer modelo pero con otra arquitectura de instrucciones. Al involucrarse el MIT en el programa Apolo tomaron al Modelo 3 como base para crear la computadora del Módulo de Comando, cuya primera versión se llamó AGC3 y tenía 12288 palabras de ROM y 1024 de RAM. Todavía no estaba decidido si iba a existir otra nave distinta para descender a la Luna y llegado el momento se definiría como adaptarle la computadora. Para noviembre de 1962 ya estaba funcionando el primer prototipo de la AGC3 que ocupaba varios armarios enteros y estaba armada con circuitos de lógica magnética en lugar de transistores. Cuando recibieron el visto bueno para probar el diseño con circuitos integrados, fabricaron un prototipo similar al AGC3 pero con estas compuertas lógicas y así demostraron que el mismo diseño podía funcionar con esta tecnología. La ventaja de la lógica magnética era su bajo consumo, por eso al pasarse a circuitos integrados el consumo se duplicó pero tanto el tamaño como el peso se redujeron y la velocidad ya no estaba tan limitada y había un mayor margen para mejorar.

Luego, en 1963 se creó un nuevo diseño para aprovechar la nueva tecnológica y se llamó AGC4. En esta nueva arquitectura los ciclos de memoria básicos, los que definen la velocidad de procesamiento de cada instrucción, dejaron de ser de 20 µs y pasaron a ser de 12 µs, lo que produjo inmediatamente una gran mejora en rendimiento [14]. También se agregó la instrucción INDEX que, además de indexar la dirección de memoria de la siguiente instrucción, también lo hacía con el código de operación y esto servía para expandir el set de instrucciones de forma similar a la instrucción EXTEND (todavía no disponible) y las instrucciones dejaron de ser sólo 8 y pasaron a ser 11. Una contra de usar INDEX para las instrucciones extendidas es que necesita dos ciclos de memoria para ejecutarse y ralentiza a todas estas instrucciones, en cambio EXTEND sólo necesita un ciclo. El tamaño de la RAM seguía siendo de 1024 por lo cual era visible en su totalidad en cualquier momento y no era necesario recurrir a un registro de selección de bancos, pero la ROM era de 24576 palabras y sí estaba dividida en bloques de 1024 palabras. Para duplicar el tamaño de la ROM usando la misma cantidad de núcleos se empezó a guardar 128 bits en cada uno en lugar de 64, lo que significa que se pasaban el doble de cables de señal por cada núcleo.

De este último diseño fue que se produjeron las computadoras Block I con un tamaño reducido para que entraran en la nave espacial. Existieron varias versiones de la Block I, con una arquitectura similar pero con cambios en aspectos como la protección de interferencias electromagnéticas. Inicialmente se pensó en hacer más modular la computadora y que los astronautas tuvieran posibilidad de realizar algún mantenimiento en caso de error. Se estudió llevar un repuesto de cada módulo electrónico para reemplazar en caso de una falla, pero luego se descartó, por un lado para no complicar la tarea y preparación ya muy

compleja de los tripulantes y por el otro porque durante la misión Mercury-Atlas 9 en 1963 ocurrieron varios problemas en la electrónica de a bordo al estar los contactos eléctricos expuestos a la atmósfera con humedad dentro de la cápsula espacial. Ahora la computadora debía estar sellada para evitar estos tipos de problemas, y al quedar todos los circuitos dentro de un bloque fijo ya no habría partes movibles ni contactos que se pudieran desconectar.

Cuando se definió la necesidad de crear un Módulo Lunar y se asignó la tarea a Grumman, este pidió que la computadora debía hacerse más liviana porque el peso de esta nave debía ser muy bajo y había que reducirlo en todos los componentes. Se pensó en ahorrar peso usando una ROM más pequeña para el Módulo Lunar, pero los requerimientos de software no eran menores y seguían creciendo y no sería posible esta reducción. Fue aquí que se hizo un nuevo modelo de la computadora con los chips flatpak que permitían ahorrar mucho peso. A la vez surgió otro requerimiento: el sistema de navegación de la Block I ayudaría a los tripulantes a orientarse en el espacio y les indicaría cómo realizar las maniobras pero no controlaría a la nave en sí, sino que de esto se encargarían los astronautas y otros sistemas electrónicos analógicos a cargo de los fabricantes de las naves. En 1964, luego de una reevaluación de los controles de vuelo, la NASA decidió que la computadora debía ocuparse del control de la nave y sus motores para mantenerla orientada correctamente. Seguiría existiendo la posibilidad de controlar manualmente la orientación y también el sistema electrónico dedicado a esta tarea, pero la computadora debería ser el sistema principal para este fin y debería poder leer las palancas de mando de la nave y estar conectada a todos sus motores. Se tenía que implementar en software un piloto automático digital para que controlara los motores utilizando reglas precisas para tener un manejo más eficiente de la nave. La computadora debería, para poder realizar todas las tareas extras, tener más memoria para incluir más programas y debería mejorar su rendimiento para realizar una mayor cantidad de operaciones complejas a la vez.

La memoria ROM creció hasta las 36896 palabras finales, pasando 192 cables de lectura por cada núcleo en lugar de 128, y la RAM se duplicó a 2048. Aprovechando el menor tamaño de las compuertas lógicas, se agregaron una mayor cantidad al diseño y esto permitió crear más instrucciones (34 en total) y mejorar la lógica de algunas ya implementadas para hacerlas más veloces. En esta versión se crearon los canales de entradas / salidas y las 7 instrucciones para trabajar sobre ellos, porque en las versiones anteriores no eran un espacio de direccionamiento aparte sino que los periféricos estaban mapeados en direcciones especiales de memoria y se usaban las mismas instrucciones que para el resto de las operaciones. Estas nuevas instrucciones de canales permitían un control más específico de bits en cada canal utilizando operaciones como AND, OR y XOR mientras se escribía o leía el contenido de un canal.

En el AGC3 se tenía como instrucciones aritmético-lógicas una para sumar (AD), otra para cambiar de signo (CS) y una para multiplicar (MP). En la Block I agregaron tres instrucciones que no estaban en los modelos anteriores: el AND binario (MASK), la resta (SU) y la división (DV). Era tan limitada la cantidad de instrucciones que no existía la instrucción

CA y no había forma de cargar en A un valor sacado de cualquier dirección de memoria. Si el valor a obtener estaba en ROM se podía usar XCH para intercambiar A con la memoria porque de todas formas la memoria no se iba a alterar pero en RAM la alternativa era usar CS y cargar en A el valor negado seguido de CS A para que lo volviera a negar antes de utilizar. En el Block II se agregó CA, también la instrucción para sumar directo a memoria (ADS) y las instrucciones de suma y resta con números de doble precisión para poder realizar estas operaciones en una sola instrucción, porque realizar esta misma tarea por software requería muchos pasos.

Cuando se incluyó la instrucción de multiplicación (MP), en el Modelo 3, en cada ciclo de memoria giraba un bit y realizaba una suma si hacía falta. Como un ciclo de memoria era muy lento (20 µs en los modelos con lógica magnética y 12 µs en el resto) la instrucción era más rápida que la implementación en software pero igual consumía muchos ciclos y tardaba más de 300 µs. En las versiones con circuitos integrados se podían realizar 12 pasos intermedios por cada ciclo de memoria de 12 µs, entonces ahí se aceleró notablemente la ejecución de esta instrucción. Como en el Block I seguía siendo lenta (94 µs) en el Block II se agregaron más circuitos para poder realizar la multiplicación en menos pasos, procesando de a dos bits por vez hasta necesitar apenas tres ciclos (35 µs). La instrucción de división (DV) también se optimizó en el Block II para poder realizarse en sólo 6 ciclos de memoria (70 µs) mejorando los 16 ciclos (187 µs) del Block I.

**Periféricos**

La principal función de esta computadora es la de ayudar en la navegación y el control de la nave, y por eso necesita interactuar con varios periféricos externos por medio de los canales de entradas / salida y los contadores. Las dos versiones de la computadora, la que está en el Módulo de Comando y la que está en el Módulo Lunar, disponen de periféricos distintos aunque gran parte de ellos son comunes iguales en ambas naves. Existían una gran cantidad de dispositivos y sensores conectados a la computadora, por ejemplo para que supiera si el Módulo Lunar y el Módulo de Comando estaban unidos o una señal que se recibía de la torre de lanzamiento para anunciar que se inició el despegue, pero los principales periféricos son la terminal de usuario, el sistema inercial para navegación, los motores de la nave, el sextante, los radares y el sistema de comunicación por radio.

**Interfaz con el usuario: DSKY**

La interfaz con la que el usuario comandaba la computadora se llamaba DSKY: un acrónimo de visor y teclado (Display y Keyboard) que eran básicamente sus componentes. Para ingresar información contaba con un teclado con 19 teclas de grandes dimensiones que podía ser usado con los guantes del traje presurizado. Cada vez que se presionaba una tecla la electrónica del teclado armaba un valor de 5 bits indicando cuál tecla se presionó y este valor era visible desde la computadora a través de un canal de entrada y a la vez se

disparaba una interrupción, lo que ahorraba al software tener que escanear el teclado periódicamente.

Se mostraba información en la terminal por medio de seis visores numéricos: tres de 2 dígitos y otros tres de 5 dígitos, donde cada dígito estaban formados por un display de 7 segmentos de color verde. Este es uno de los primeros usos de este tipo de visores, que luego se harían populares en la década del 1970. Anteriormente se recurría a distintas tecnologías para mostrar números, como tubos Nixie, que representan en una válvula de vacío uno de diez posibles dígitos, o medios electromecánicos como agujas o ruedas con dígitos numéricos. Estos visores no eran LEDs, como los actuales, porque estos todavía no eran prácticos, sino que eran electroluminiscentes, la tecnología comúnmente utilizada en la iluminación de fondo de los relojes de cristal líquido, aunque en este caso trabajaban con una tensión alta de 250 V (Figura 10).

Cuando se cuenta con varios visores de 7 segmentos se recurre a un conexionado en forma de grilla para poder controlar todos los dígitos con la menor cantidad posible de señales, enviando la información de un visor por vez mientras se indica a cuál de ellos corresponde. Lo común es valerse del efecto de persistencia de la visión y mostrar la información varias veces por segundo pero sólo un dígito por vez, lo que le produce al observador la sensación de estar viendo todos los dígitos juntos. Aquí no se usa esa técnica porque desperdiciaría un gran porcentaje del tiempo de este procesador tan lento en ese refresco de visores, por eso cada vez que se le envía un valor a un dígito este recuerda el valor y lo mostrará de forma continua sin necesidad de refrescos por parte de la computadora. La información de cada dígito es almacenada en un relé de enclavamiento o biestable que siempre mantiene el último valor que se le escribió desde de la computadora [3] y cada vez que lo que muestra un visor cambia se escucha el ruido del relé moviéndose.

Por software se determina que mostrar en cada visor y los visores de dos dígitos se reservan para información técnica del estado del software, especialmente de la interfaz de usuario. En los visores de 5 dígitos se muestran cantidades grandes que utiliza el programa, por ejemplo una posición en el espacio o una velocidad, aunque no cuentan con un punto decimal para mostrar valores fraccionados sino que es el operador quién debe conocer en donde está el punto. Por ejemplo si se muestra 12.456 se lo muestra como 12456 y el operador debe saber, según el programa que está usando, que es una cantidad con 3 decimales. Una característica llamativa de estos tres visores es que, además del signo – que son capaces de mostrar a la izquierda, también pueden mostrar el signo *+*, y entonces cuando muestran una cantidad siempre se especifica el signo: por ejemplo si la velocidad es negativa muestran –12456, pero si es positiva no muestran 12456 sino +12456.

Esta terminal se ubica junto con la computadora en el panel frente al puesto del piloto de esa nave. En el Módulo de Comando, como el sextante se tuvo que ubicar en una posición que no era accesible al piloto cuando estaba en su puesto, se agregó una segunda terminal junto al sextante y de esta forma se podía operar la computadora también mientras se utilizaba ese instrumento [2].

Además de los visores numéricos se cuenta con varias luces incandescentes en un panel donde cada una indica un estado. Fuera del panel existe una alarma de error conectada a distintos sistemas electrónicos de la nave y que se activa cuando se detecta una condición errónea para alertar a los tripulantes de que algo no está en el estado correcto. Al activarse se encienden luces que indican la causa de la alarma y también emite una señal sonora para captar la atención inmediata de los astronautas, que deben apagarla manualmente. La computadora también está conectada a este sistema de alarma y desde el software se puede encender la alarma para alertar de un problema en la ejecución del programa. Esta alarma no indica de ninguna forma cuál fue la causa del error de software sino que los astronautas, una vez avisados, deben consultarle el código a la computadora ingresando un comando en el teclado.

**Figura 10.** Terminal DSKY: visor y teclado de la computadora.

En el Módulo Lunar también se incluyen otros indicadores específicos comandados por la computadora para proveer de información rápida a los astronautas mientras alunizan. El indicador de altitud es una cinta que muestra la altitud a la que la computadora cree estar y también la velocidad de descenso. Existe un indicador de 2 dimensiones de velocidad en

los ejes x e y donde la computadora envía la velocidad calculada con respecto a la superficie lunar y que se debe reducir lo más posible antes de tocar la Luna para no destruir la nave.

**Palancas de mando**

Para controlar el movimiento de la nave existen dos palancas: una para controlar la rotación y otra para desplazamiento. La palanca de rotación se controla con la mano derecha (por su posición en el puesto de comando) y se puede rotar en tres ejes para poder controlar el giro en cualquier dirección. La palanca de desplazamiento se controla con la mano izquierda y tiene forma de T. Con este control, que también es de tres ejes, se controla la aceleración que gana la nave hacía cualquier dirección del espacio. Con la palanca de rotación se controlan disparos de los motores de giro de a pares ubicados a ambos lados de la nave y disparando en direcciones opuestas para producir un giro sobre su centro de masa. La palanca de desplazamiento controla los mismos tipos de motores, pero usándolos de a pares que apuntan en la misma dirección para que la nave gane velocidad en un eje y no rote. Estas palancas son digitales: envían una señal de contacto cerrado o abierto en cada dirección sobre la que se apunte sin indicar cuanto está desplazado (seis señales en total). Existe una excepción y es la palanca de rotación del Módulo Lunar que es analógica y envía el ángulo en que está en cada uno de sus tres ejes para tener un control más preciso durante el alunizaje. La conversión de analógico a digital de esta palanca es más simple que la de los otros periféricos porque no requiere mucha precisión (valores de 0 a 31) ni mucho muestreo (hasta 33 veces por segundo) y consiste en el envío de una señal con modulación de ancho de pulso para cada uno de los 3 ejes. La computadora lee la posición de cada eje iniciando la carga de un capacitor y midiendo cuánto tarda en descargarse. En el Módulo Lunar también hay un tercer control, pero de un solo eje, que sirve para regular la velocidad de descenso cuando la nave está cerca del suelo.

La información de estos controles se envía a la computadora para que el programa decida, según la configuración del software, de qué forma enviar disparos a los motores para realizar los movimientos deseados sobre la nave. De todas formas, estos mandos no son exclusivos de la computadora sino que se pueden usar para controlar la nave pasando por alto este sistema de navegación: dependiendo de cómo estén configurados algunos interruptores, las señales de estos controles irán hacia un sistema secundario de estabilización (que en el Módulo de Comando es totalmente distinto al del Módulo Lunar) o se pueden usar para encender y apagar directamente los motores de orientación, lo cual no es muy óptimo porque se desperdicia mucho combustible al manejarlos de esta forma tan imprecisa.

Con estas palancas no se maneja el motor principal de la nave, sino que están para las maniobras más finas, no para cuando hay que hacer una maniobra orbital con un gran cambio de velocidad.

## Unidad inercial

Este instrumento, que es la base de todo el sistema de navegación, le dice a la computadora la orientación de la nave en cada momento y la aceleración que ocurre en cada eje para que calcule todas las velocidades y mida las fuerzas que actúan sobre ella. Visto de afuera es una esfera de unos 27 cm de diámetro, lo que representa una gran miniaturización teniendo en cuenta la cantidad de componentes mecánicos movibles e instrumentos electromecánicos que incluye y que entregan mediciones muy precisas [2] (Figura 11).

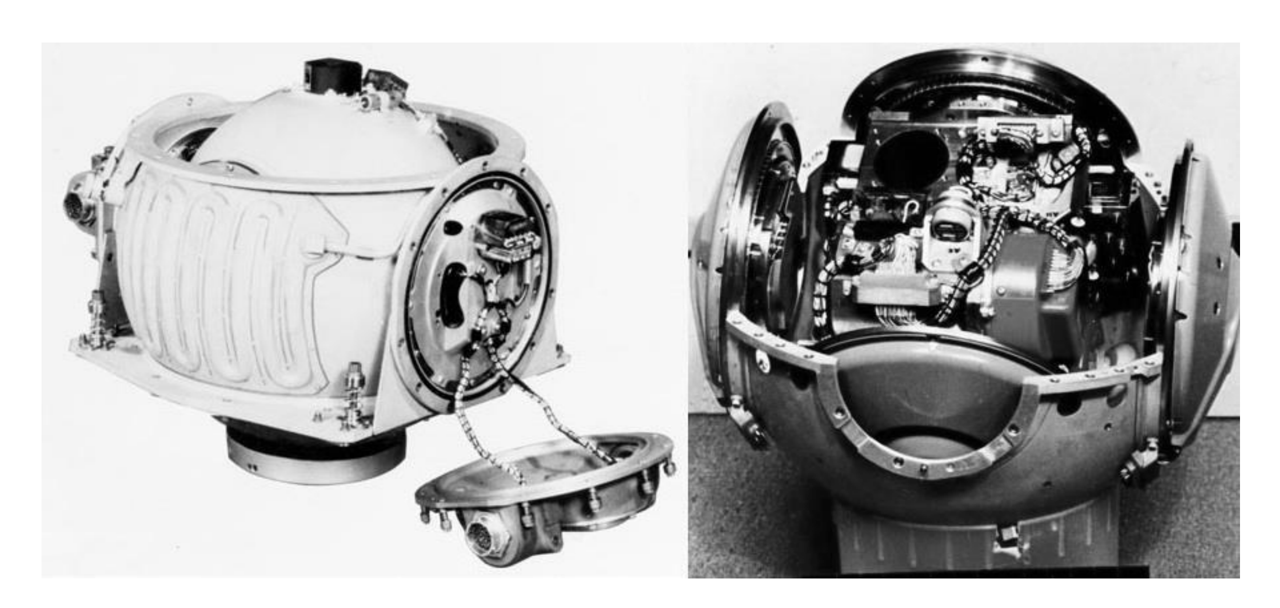


**Figura 11.** Unidad inercial usada en las naves Apolo. A la izquierda se ve una unidad cerrada y a la derecha una abierta con los componentes internos a la vista.

La unidad inercial está fijada a la nave y consta de tres cardanes (o gimbals en inglés) de distinto tamaño, uno dentro del otro, que contienen a la plataforma estable en el centro y se unen entre sí con ejes que pueden girar usando un motor eléctrico. La plataforma estable tiene tres giroscopios orientados cada uno en un eje distinto, todos perpendiculares entre sí que envían señales a los motores de los tres ejes para mantener a la plataforma siempre alineada con los giroscopios, entonces cuando la nave cambia su orientación, los motores compensan esta rotación para que la plataforma siga apuntando en la misma dirección.

En cada uno de los tres ejes movibles hay un resolver, un instrumento que genera pulsos con cada pequeño giro del eje y le envía esta información a un contador de la computadora, que en todo momento conoce los tres ángulos que establecen la orientación de la nave con respecto a la plataforma estable. La computadora puede controlar cada uno de los giroscopios para que se incline para uno u otro lado (movimiento de precesión) para poder reorientar la plataforma estable en cualquier dirección en caso de necesitar calibrar a la unidad inercial. Si tomamos a la nave como un vector en la dirección en que apunta el motor principal, entonces para indicar su orientación en el espacio tridimensional se

necesitarían dos ángulos, como cuando se utilizan coordenadas polares en tres dimensiones. Pero como la nave no es un objeto puntual, no alcanza con saber en qué dirección apunta sino que se necesita un ángulo más para saber hacia dónde apuntan los lados de la nave y así determinar cómo usar los propulsores que posee alrededor, por eso necesita estos tres ángulos que le provee la unidad inercial para conocer su orientación en el espacio.

Dentro de la plataforma estable existen tres acelerómetros que apuntan cada uno en direcciones distintas para medir la aceleración en los tres ejes. Una aplicación posible es enviar a la computadora todo el tiempo las aceleraciones medidas y que la computadora con esta información la integre y obtenga la velocidad, pero para hacer esto es necesario una frecuencia de muestreo muy importante porque si se dejara de tener en cuenta una aceleración, por más breve que fuera, afectaría todas las cuentas de velocidad de ahí en adelante. Esta computadora no cuenta con una capacidad de procesamiento muy alta como para estar atenta a cambios bruscos pero instantáneos que se pueden producir, por ejemplo, al encender un motor, entonces lo que se hace es integrar estas aceleraciones por afuera de la computadora. La interfaz electromecánica entre los acelerómetros y la computadora calcula todo el tiempo la velocidad y es este cambio en la velocidad lo que le envía a un contador de la computadora. Entonces el software puede leer el contador de un acelerómetro cada 2 segundos y lo que obtendrá es el cambio de velocidad ocurrido en ese lapso de tiempo, sin importar cómo se produjo, si con un pequeño impuso de unos pocos milisegundos o con un empuje constante de 2 segundos. En cada lectura borra el valor del contador para que la siguiente lectura arroje la diferencia de velocidad con lo leído anteriormente.

**Indicador de actitud**

En el panel de instrumentos existe una bola que rota en tres ejes que sirve para indicar a los tripulantes la orientación de la nave con respecto a la unidad inercial. Este instrumento, llamado FDAI (Fligh Director Attitude Indicator), es similar al horizonte artificial utilizado en los aviones pero en este caso se alinea con las estrellas. Este indicador no es controlado por la computadora sino que se mueve según la orientación de la unidad inercial porque está conectado directamente a sus señales de salida (Figura 12).

Al diseñar el sistema de navegación y el instrumental de abordo, se lo hizo de forma científica y no se tuvo en cuenta la enorme experiencia de los astronautas como pilotos de aviones, en donde siempre contaban con un horizonte artificial que mostraba la actitud con respecto al suelo. Cuando la nave orbita la Tierra o la Luna, la unidad inercial permanece apuntando siempre hacia la misma dirección en el espacio, por ejemplo a una estrella en particular, pero la nave generalmente va rotando para apuntar siempre hacia abajo en el planeta, por ejemplo para poder observar continuamente la superficie. La reorientación de la nave para seguir al suelo la realiza el software de la computadora, pero el indicador de actitud, al rotar la nave, gira en relación a ella indicando siempre la orientación de la unidad

inercial. Esto a los astronautas no les servía porque era confuso tener un indicador que no quedara fijo con el horizonte en un vuelo en que el ángulo de la nave sí está fijo con este.

No existía la posibilidad de corregirlo en el software y modificar la electrónica de la nave para agregarle un nuevo modo no era sencillo. La solución de compromiso que se encontró fue llevar una cajita extra para conectar al panel, que le envía una señal periódica a unos de los motores que gira un eje de la bola para que esos pulsos estén sincronizados con la velocidad de rotación de la nave y de esta forma se hace que la posición de este indicador quede fija con el horizonte. Esta caja se llamó ORDEAL [12], acrónimo de Orbital Rate Drive Electronics for Apollo and LM o electrónica de control de frecuencia de órbita para Apolo y Módulo Lunar, pero ese nombre también significa sufrimiento en inglés.

**Figura 12.** FDAI indicador de actitud de la nave.

## Gimbal Lock

Una limitación importante de la unidad inercial es la existencia de un estado llamado *Gimbal Lock* (bloqueo de cardán). Tan nefasto es este estado que la tripulación debía estar siempre atenta para evitarlo y en el panel se tienen indicadores para alertar cuando se aproximan a él.

La medición de la orientación se puede hacer porque la plataforma estable rota de forma independiente al resto de la nave, por eso se la puede usar como referencia en cada momento. Para que se pueda mover libremente se cuentan con los tres cardanes (gimbals) que dan tres grados de libertad para que con cualquier reorientación de la nave roten los ejes correspondientes. Si la nave se reorienta de la forma que fuera, ese movimiento se puede expresar con tres ángulos y como la plataforma se mueve en esos tres ángulos la nave y la plataforma estable se pueden orientar de forma independiente.

El problema surge cuando se pierde un grado de libertad: si la plataforma queda unida a la nave con sólo dos ejes movibles cualquier movimiento de la nave que no se pueda

representar con esos dos ejes hará que la plataforma gire junto con la nave y de esta forma deja de ser una referencia útil.
Por la forma en que está armada esta unidad inercial, existen orientaciones en que dos ejes quedan apuntando en línea, entonces ambos ejes se combinan formando uno solo y esto equivale a perder un grado de libertad. Entre el eje α y el eje β (Figura 13.a) hay siempre 90º al estar montados los dos sobre el mismo cardán. Lo mismo ocurre entre el eje β y el eje θ que también comparten cardán separados a 90º. El problema es que entre los ejes α y θ la orientación es variable porque están ubicados en distintos cardanes que se pueden orientar de variadas formas. Si la nave se orienta de forma tal que estos dos ejes se alineen (Figura 13.b) entonces aquí ocurrirá el Gimbal Lock.

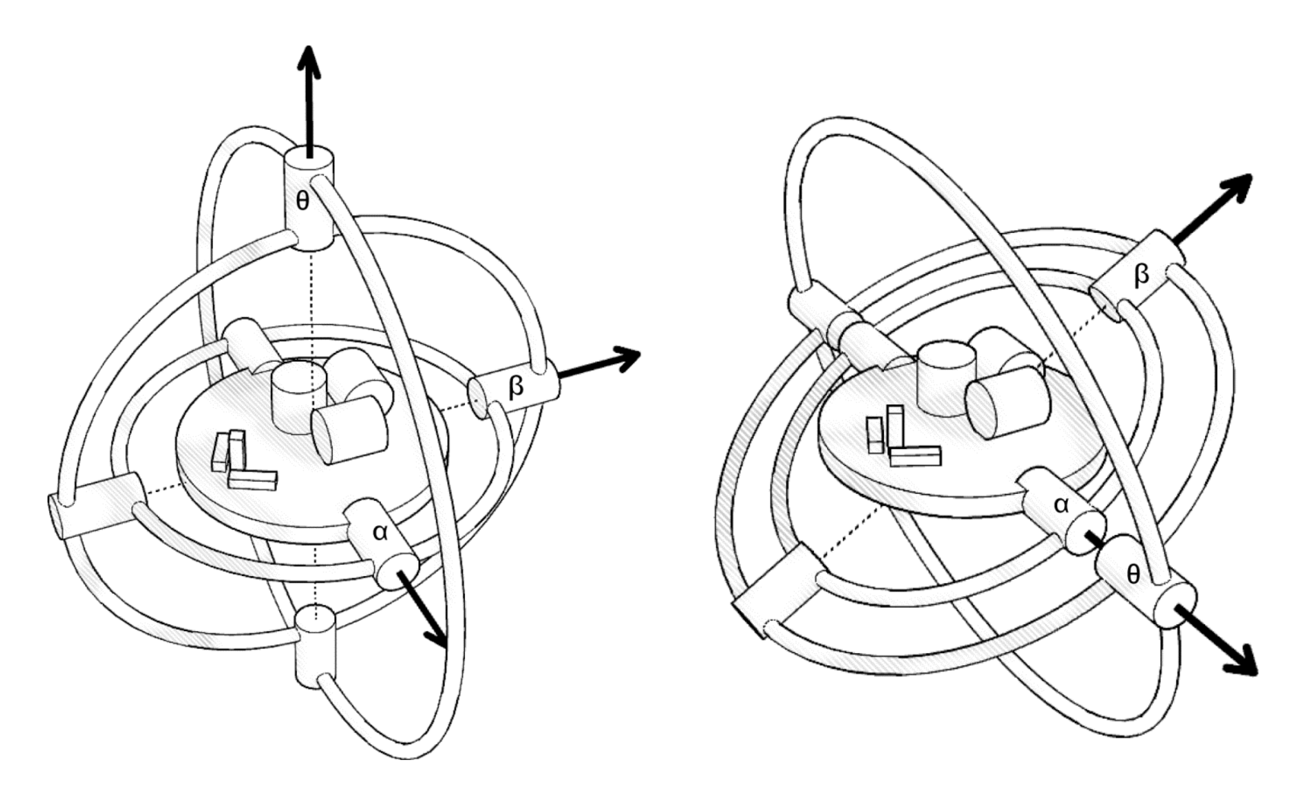



**Figura 13.** A la izquierda la unidad inercial con tres ejes independientes. En la izquierda un ejemplo de Gimbal Lock: el eje α y el θ ahora son el mismo entonces hay un grado de libertad menos.

Este problema era conocido y también lo era su solución: agregar un cardán extra. Este cardán era controlado por la electrónica justamente para evitar que ocurriera una alineación entre dos ejes. La computadora luego tendría que leer cuatro ángulos en lugar de tres y convertirlos matemáticamente a un sistema de tres ángulos. Grumman, el fabricante del Módulo Lunar, insistió en agregar este cardán para evitar el problema de Gimbal Lock que ya era conocido de varios sistemas misilísticos. En el MIT analizaron que el trayecto ida y vuelta a la Luna transcurría básicamente en un plano (el plano en que la Luna gira alrededor de la Tierra) y de esta forma la nave no necesitaba poder orientarse en cualquier dirección, entonces optaron por usar sólo 3 cardanes para ahorrar peso y espacio en la nave. Hubo una disputa que se terminó definiendo a favor del MIT, pero con el tiempo ellos también consideraron que fue una mala decisión porque para evitar este estado de bloqueo se

requirió mucho entrenamiento y esfuerzo durante las misiones, además de instrumentos en el panel y rutinas en la computadora [2].

**Problema con la unidad inercial y el sistema complemento a 1**

Un problema que surgió con los sensores de la plataforma inercial era que inicialmente a los fabricantes del instrumento se les indicó que el contador que usaba la computadora para representar el ángulo de la plataforma era de 15 bits, y como ellos debían generar los pulsos de incremento y decremento del contador para indicar a la computadora el ángulo en que se encuentra, supusieron que en un giro completo en un eje del instrumento debería arrojar 32768 ($2^{15}$) pulsos. Para que el sistema electromecánico (el resolver de cada eje) arroje esa cantidad de pulsos con un giro completo se necesitaría una rueda con 32768 terminales eléctricos que en una vuelta completa genere esa cantidad de pulsos, o en un giro menor una cantidad proporcional. Para simplificar el mecanismo se recurre a combinaciones de engranajes para que la rueda con los terminales sea más pequeña pero realice mayor cantidad de giros por vuelta completa pero arrojando la misma cantidad de señales. Un engranaje que es movido por otro girará a distinta velocidad según la relación de la cantidad de dientes que tienen entre ambos engranajes, por ejemplo si un engranaje de 30 dientes mueve a otro que tiene 10, este último dará tres vueltas por cada una que realice el primero y así se puede multiplicar la cantidad de vueltas que da un eje con respecto al primero. Se podría, por poner un ejemplo, tener una serie de engranajes que produzcan que el último de 32768 vueltas por cada una que da el primero y genere esa cantidad de pulsos en una vuelta del eje principal. Como en este caso se trabaja con un número que es una potencia de 2 ($2^{15}$), también es múltiplo de 2 y todas sus potencias hasta $2^{14}$, se tiene una gran cantidad de valores para seleccionar engranajes para obtener esa cantidad de pulsos en un giro completo.

El inconveniente con este diseño fue que no se tuvo en cuenta que la computadora usaba complemento a 1, y en este sistema el 0 está repetido y la instrucción de incremento o decremento ignora uno de los dos ceros (dependiendo del sentido de la operación) por lo que en realidad hay sólo 32767 valores posibles en un giro completo y esa es la cantidad de pulsos que debería generar el instrumento en una vuelta completa [6].

En las primeras pruebas en tierra, estando la unidad inercial completamente quieta, surgió que cada 24 horas el sistema se desfasaba en 1 (0,01 grados) con el ángulo real de la plataforma inercial, al realizar la Tierra un giro completo sobre su eje. Si bien esta diferencia es muy chica, en un vuelo donde se está permanentemente rotando sobre su eje para recibir de forma pareja el calor del Sol, este corrimiento de 1/32767 por cada giro en poco tiempo produce un corrimiento importante que afecta a todo el sistema inercial.

Mecánicamente no era simple corregirlo porque 32767, a diferencia de 32768, tiene sólo tres factores (7,31 y 151) y la elección de engranajes se hacía muy complicada. Por esto se modificó la electrónica de la computadora para que los contadores que se usaban para obtener los distintos ángulos del sistema inercial realizaran la aritmética usando el sistema

de complemento a 2, y básicamente son los mismos pasos salvo que no se realiza el acarreo al final en caso de cambio de signo, así que ahora existe una sola representación del 0 y no se saltea ninguna combinación de 15 bits. También se debieron agregar instrucciones para realizar restas de valores en complemento a 2 para poder comparar los valores leídos aquí sin interpretarlos mal como si fueran de complemento a 1.

**Control de los motores**

Cada nave tiene dos tipos de motores. El principal, que es el que la acelera lo suficiente para hacerla desplazar grandes distancias por el espacio, y luego están los pequeños propulsores ubicados a los costados que se encienden de a pares para orientar a la nave. La computadora tiene control de todos los motores y es capaz de encenderlos y apagarlos en cualquier momento para orientar y desplazar la nave en el espacio. Cada motor es controlado con solo un bit de un canal de salida y la aceleración total que aplicará a la nave dependerá del tiempo que permanezca encendido ese bit cuya responsabilidad de administrarlo correctamente corresponde al software. Los astronautas disponen de una gran cantidad de interruptores eléctricos y válvulas de combustible para habilitar el control de los motores por parte de la computadora, de esta forma en caso de emergencia pueden controlarlos manualmente ignorando al piloto automático [12]. También se deja desactivado el control de la computadora de los motores en los momentos que no es necesario usarlos para evitar su uso por error.

Para desplazar la nave, como se vio anteriormente, alcanza con primero orientarla en dirección en que se piensa viajar (utilizando los propulsores de orientación) y luego se enciende el motor principal que acelera a la nave en esa dirección. Si bien esto es correcto, es una simplificación porque se está suponiendo que la nave es un objeto puntual o que el motor está perfectamente alineado con el centro de masa de la nave. El centro de masa es el punto donde si se le aplica una fuerza se puede considerar que toda la masa de la nave está concentrada en ese solo punto. Si en cambio se le aplica una fuerza que no pasa sobre el centro de masa, una parte de la fuerza, en lugar de acelerar a la nave en la dirección buscada, la hará girar sobre su centro de masa, desperdiciando energía y reorientando el motor en una dirección errónea (Figura 14).

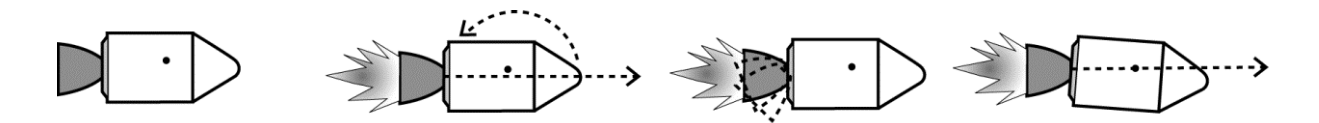


**Figura 14.** Nave con el centro de masa corrido de eje central. En la segunda imagen cuando el motor empuja no lo hace sobre el centro de masa y eso hace que la nave gire sobre sí misma. En la imagen siguiente se ajusta el ángulo del motor y en la última se ve como empuja sobre el centro de masa y toda la fuerza se aprovecha para hacer avanzar a la nave.

Para poder aplicar la fuerza sobre el centro de masa se monta al motor principal en una estructura movible que es orientable por medio de dos motores eléctricos, entonces la computadora gira al propulsor hasta que coincida con el centro de masa de la nave. El software debe verificar al encender un motor que la aceleración se aplique sobre el centro de masa, sino ajusta el ángulo del motor hasta encontrar el punto ideal.

**Sextante**

El sextante es el instrumento óptico usado en el Módulo de Comando para medir con precisión ángulos de cuerpos celestes con respecto al punto de observación y es usado para calibrar la orientación de la nave con respecto a las estrellas fijas y su posición en el espacio.

Consta de dos visores independientes [12], uno es fijo llamado visor de paisaje (Landmark Line of Sight) y para apuntar con él hacia un objetivo es necesario orientar la nave entera. El otro visor es movible en un solo eje, y es conocido como visor de estrellas (Star Line of Sight) y se puede apuntar con precisión moviéndolo hasta 57 grados. Lo captado por ambos visores es combinado para formar una sola imagen que es lo que ve el astronauta. El sextante completo con ambos visores podía ser girado sobre su centro para seleccionar el eje en que se ajusta el visor movible sin tener que rotar toda la nave (Figura 15).

Ambos visores del sextante tienen un aumento de 28x para poder medir los ángulos con la mayor precisión posible pero, a causa de esto, su campo de visión es de apenas 1,8 grados. Para ayudar a orientar la nave buscando las estrellas, junto al sextante hay un telescopio de búsqueda que apunta hacia la misma dirección que el visor fijo del sextante. Este telescopio es simplemente un visor que no tiene ningún aumento y pero que posee un campo de visión mucho más amplio y por eso es útil para ubicar una estrella en el espacio. Una vez que se sabe hacia dónde apuntar para observar la estrella la ubicación más precisa se realiza con el sextante.

El sextante está conectado con la computadora y eso permite que esta conozca con precisión los dos ángulos en que está orientado el instrumento. Además de esto, el sextante tiene un botón, llamado MARK, que le envía una señal a la computadora cuando se presiona, para que entonces esta tome la medición del sextante en ese preciso instante. Esto es útil porque no sólo importa la orientación del sextante (y la nave) sino el momento exacto en que se realiza la observación, porque, como la nave está en movimiento, lo que se ve también va cambiando. El software toma los ángulos del sextante y la hora en que se realizó la medición y con esos datos puede calcular la posición de la nave al momento de realizar la observación.

**Radares**

El sistema de navegación del Módulo Lunar, al calcular su posición en el espacio con respecto a la Luna, debería ser capaz, en teoría, de realizar el alunizaje con sólo esta información. La realidad es que se conocen las coordenadas de la nave con respecto al

centro de la Luna, pero no en relación con el lugar en la superficie donde se descenderá, y esto es lo que se necesita saber para descender con suavidad. Los descensos se planificaban con información obtenida de misiones Apolo previas o sondas robóticas enviadas para fotografiar la superficie, pero la altura de un punto en el terreno (la referencia es la distancia al centro de la Luna porque esta no cuenta con océanos para tomar el nivel del mar) se estimaba a grandes rasgos y no había una certeza de a qué altura se detendría el descenso. De todas formas, la información del sistema inercial es calculada de forma indirecta y una pequeña desviación de apenas unos metros puede hacer que la nave choque contra el suelo al estimar mal su posición, por eso se necesita una medición directa de la altura para hacer contacto con la Luna con suavidad.

**Figura 15.** Sextante y telescopio de prueba. El astronauta Ron Evans, de la misión Apolo 17, realiza un ensayo en el simulador del Módulo de Comando.

Para contar con una referencia externa de posición, el Módulo Lunar posee un radar que mide distancia y velocidad de aproximación a la superficie lunar que obtiene mediciones cuando se encuentra a distancias menores a 10 km. Esta información se lee desde la computadora y esta usa la información, junto con las coordenadas del sistema de navegación, para realizar el alunizaje.

Esta nave, además, cuenta con un radar de aproximación al Módulo de Comando que transmite una señal que es repetida por un transceiver en esta otra nave y se orienta automáticamente para obtener la distancia y dirección a la que se halla. La información de ángulo y dirección es usada por la computadora en la maniobra de aproximación entre las naves.

**Comunicación por radio**

Como los controladores de vuelo que supervisan el funcionamiento de la nave a miles de kilómetros de distancia necesitan que esta les envíe constantemente información acerca de su estado, lo que se conoce como telemetría, en cada nave existe un sistema electrónico que transmite una señal de radio donde se informa lo leído de los muchos sensores que están repartidos por todo el vehículo. Este sistema no depende de la computadora sino que es un circuito electrónico dedicado exclusivamente a la telemetría [3] y la señal que genera es digital y transmite de forma serial, un bit por vez, el estado de, por ejemplo, todos los interruptores controlados por los astronautas y las válvulas que se abren o cierran tanto de forma manual como a través de algún sistema electrónico. La velocidad de transmisión varía según que antena se esté utilizando para emitir la señal: con la antena parabólica de alta ganancia es posible transmitir datos seriales en el modo de alta velocidad que es de 51 Kb/s, en cambio, cuando se utiliza la antena omnidireccional la velocidad es de apenas 1,6 Kb/s. Este sistema de transmisión envía, además de los bits con la información útil, otros que le dan formato a los paquetes de datos: bits para indicar cuando comienza y termina el paquete y bits que identifican el tipo de información que contiene. Es necesario que el que reciba la información pueda descartar los paquetes que contienen información errónea a causa de problemas en la comunicación, por eso se envía en cada paquete bits redundantes de paridad para que el que procese la información pueda validar que lo leído sea correcto. Como este formato de transmisión no es generado por una computadora, no se pueden utilizar algoritmos avanzados que den una mayor garantía de que los datos sean válidos y generados por la nave real, y mucho menos para encriptarlos para que nadie más pudiera generar señales falsas o hacer espionaje, pero aun así cumplía su objetivo en una época en que no habían muchas señales de radio digitales que se pudieran confundir con esta.

Para poder enviar valores tomados de señales analógicas, por ejemplo termómetros, sensores de presión o de consumo eléctrico, se dispone en la nave de un conversor analógico-digital electrónico con un multiplexor de entradas que toma señales de a una a la vez y genera un valor de 8 bits que se incluye dentro del paquete con datos digitales. Este conversor de señales analógicas es grande y pesado pero sólo es necesario uno para toda la nave, a diferencia de los conversores que utiliza la computadora que, además de necesitar mayor precisión, también debe existir uno por cada señal al tomar muestreos más grandes. El sistema de transmisión, siguiendo una secuencia predefinida, selecciona qué entrada analógica tomar para transmitir esos 8 bits en cada paquete de datos. Algunas señales se

envían más seguido que otras, según el muestreo que cada una necesite para tener un mejor monitoreo remoto del estado de la nave.

Es muy importante que se pueda monitorear remotamente lo que está realizando la computadora, por eso en los paquetes con información digital también viajan datos generados por esta. Dos de los canales de salida de la computadora están conectados con este sistema de transmisión de datos de la nave y los 30 bits que se escriben ahí serán transmitidos de forma serial dentro del paquete de datos [15]. El sistema de transmisión dispara una interrupción cada vez que termina de enviar los 30 bits de la computadora para avisarle a esta que ya puede cargar las próximas dos palabras con datos para transmitir. El software procesa esta interrupción y vuelca en esos dos canales de salida la siguiente información que desea enviar y de esta forma selecciona que información transmitir para que remotamente se conozca su estado. Como esta interrupción se produce 50 veces por segundo, y, en consecuencia, la computadora puede transmitir apenas unas 100 palabras cada segundo, el programa debe ser selectivo en lo que transmite para proveer a los ingenieros de información útil actualizada. Lo que se transmite más frecuentemente es el estado del teclado y del visor, para informar de todas las operaciones realizadas por los astronautas y luego el estado del sistema de navegación: las coordenadas y la velocidad calculadas por el software. También se envía el estado de algunos canales de entrada para poder controlar su correcto funcionamiento, y, según qué programa este corriendo en ese momento, información útil para analizar los cálculos que se estén realizando. El software tiene un modo para enviar el contenido completo de la RAM, y cuando se selecciona este modo se va transmitiendo cada palabra de esta memoria en orden y así, en 20 segundos se puede enviar su contenido completo.

Este sistema de generación de datos de telemetría solamente transmite y no recibe ninguna señal externa ni espera confirmación de los datos enviados. La computadora, en cambio, puede recibir datos remotamente a través de un contador de entrada que está conectado a un receptor de radio y que se usa como entrada serial y dispara una interrupción cada vez que entran 15 bits. Esta palabra que se lee está pensada para interpretarse como un valor de 5 bits que identifica una tecla, usando el mismo código que generan las teclas del teclado real y de esta forma se reutilizan las mismas rutinas de software de entrada de comandos y lo leído por radio es apenas un teclado más. Para filtrar errores de comunicación, dentro de la palabra de 15 bits que se recibe está el mismo código repetido 3 veces y una de ellas con los bits negados. Si el software verifica que los 15 bits cumplen todos estos requisitos se toma el valor como una tecla, y en caso contrario se enciende una luz de error. Este sistema tan básico de verificación de errores es destacado en la documentación de la época [3] como una técnica muy avanzada para validar la información y lo es comparado con los paquetes de telemetría, pero aun así sería muy fácil que un tercero tomara el control de la computadora de forma remota al usar un chequeo tan básico, aunque los astronautas pueden deshabilitar la entrada de datos remotos con un interruptor.

Por medio de la utilización de los dos medios de comunicación de datos, la telemetría de la nave y la señal directo a la computadora, se puede operar este equipo remotamente cómo si se estuviera dentro de la nave. Por medio de la entrada de datos a la computadora se ingresan códigos de teclas y leyendo la información de telemetría se verifica que se leyó correctamente la entrada de teclado y se ve lo que muestra el visor en respuesta a lo ingresado. Con esta forma de trabajar no existe ninguna operación en la computadora que se pueda realizar remotamente por radio y que no se pueda ejecutar de la misma manera desde la terminal dentro de la nave.

## CONCLUSIONES

Hasta aquí vimos los componentes que formaban al sistema de navegación de las naves Apolo: la computadora en sí, la terminal, los periféricos de entrada y salida y cómo estaba integrada la computadora a las naves. En la próxima parte se estudiarán las características y desarrollo del software que corría en esta computadora.

### BIBLIOGRAFÍA


[1] P. Parker (2003). Apollo and the integrated circuit. http://history.nasa.gov/alsj/apollo-ic.html (23 de mayo de 2016).

[2] D. A. Mindell (2008). *Digital Apollo*. MIT Press.

[3] E. C. Hall (1972). Apollo guidance, navigation and control. *MIT´s Role in project Apollo,* Volume III: Computer Subsystem. Charles Stark Draper Laboratory.

[4] Texas Instruments (2007). Texas Instruments Special ICs produced for Minuteman missile. http://www.ti.com/corp/docs/company/history/timeline/defense/1960/docs/62-special_ics.htm (29 de mayo de 2016).

[5] E. Hall (1996). *Journey to the Moon: The History of the Apollo Guidance Computer*. American Institute of Aeronautics & Astronautics.

[6] R. L. Alonso (2004). Evolutionary Dead Ends. MAPLD International Conference. 8 a l 10 de septiembre de 2004.

[7] F. O'Brien (2010). *The Apollo Guidance computer Architecture and operation*. Springer Praxis.

[8] O. Nash y Ben North (2011). Proyecto para crear una memoria de núcleos magnéticos. http://www.corememoryshield.com (12 de abril de 2016).

[9] A. L. Hopkins (1967). Electronic Navigator Charts Man´s Path to the Moon. *Electronics* 109, 9 de enero de 1967..

[10] M. Stephan (2013). Making a Core Rope Read-Only Memory. http://hackaday.com/2013/10/09/making-a-core-rope-read-only-memory/ (16 de abril de 2016),

[11] N. D. C. R. Duncan Copp (dir.) (2008). *Moon Machines Episode 3. The Navigation Computer.* Science Channel.

[12] W. D. Woods (2011). *How Apollo Flew to the Moon*. Springer Praxis.

[13] MIT Science Reporter (1965). Computer for Apollo. J. Fitch (entrevista) http://mit150.mit.edu/multimedia/science-reporter%E2%80%94computer-apollo-1965 (23 de marzo de 2016.)

[14] H. Blair-Smith (1997). Annotations to Eldon Hall's Journey to the Moon. http://authors.library.caltech.edu/5456/1/hrst.mit.edu/hrs/apollo/public/blairsmith.htm (6 de junio de 2016).

[15] M. S. Johnson (1971). The software effort. *MIT's role in project Apollo, Volume V*. Charles Stark Draper Laboratory.